Ch - 1

Overview of Python and Data Structures

1. List the Advantages of Python

Ans (i) Easy Syntax
→ Pythons syntax is easy to learn
(ii) Reliability
→ Python can be used to prototype and test code which is later to be implemented in other programming languages
(iii) High-Level Language
→ Python looks more like a readable, human language than like a low-level language
→ Thus gives you the ability to program at a faster rate
(iv) Cross Platform
→ Python runs on all major operating system like Microsoft Windows, Linux and Mac OS X
(v) Widely Supported
→ Python has an active support community with many websites, mailing lists and USENET "netnews" group
(vi) It's Safe
→ Python doesn't have pointers like other C-based languages, making it much more reliable
(vii) Huge amount of additional Open-Source Libraries
→ There are over 300 standard library modules that contain modules and classes for a wide variety of programming tasks

Page No.
Date

0 Python Data Structures

→ Python Data Structures include string, array, list, tuple, set and dictionaries

(1) String

→ Strings are a collection of characters which are stored together to represent arbitary text inside a python program

→ Strings are arrays of bytes representing Unicode characters

0 Operations in String

1) Concatenation: The addition operator (+) is used to concotentation of strings

Eg:

```
First = "Sujal"
Second = "Vachhani"
First + Second " " + Second
```

Output: "Sujal Vachhani"

2) Repetition: "*" asterisk is used

Eg:

```
'A' *5
```

Output: 'AAAAA'

3) Indexing and Slicing:

Eg: what = 'Python Book'

```
what[5]
```

Output: 'n'

(ii) Array

→ Python does not have built in support for Arrays Python lists are used to serve the purpose

→ To create arrays in Python, we need to import the array module to create arrays

Example:

```
import array as ary
b = ary.array('i', [2,4,6,8,10])
print(b[2])
```

Output: 6

(iii) List

→ List provide a general mechanism for storing a collection of objects indexed by a number in Python

→ List are a mutable collection of items. We use square brackets to surround the items []

Eg:

```
numbers = [10, 11]
characters = ['a', 'b']
print(numbers, characters)
```

Output:

```
[10, 11] ['a', 'b']
```

(iv) Tuple

→ A tuple is a sequence of values. The values can be any type and are indexed by integers unlike lists.

→ Tuples are immutable

Syntax

Tuple I = (item 1, item 2, ..., item n)

Example

```
item = (0, 'Suresh', 'Sujal', '99.9')
print (item[2])
```

Output

Sy Sujal

v) Set

→ Set is an unordered collection of simple objects in Python

→ You can use curly braces to give an expression whose value is a set.

Example:

```
a = {1, 3, 2, 5, 4}
print (a)
```

Output:- {1, 2, 3, 4, 5}

(vi) Dictionary

→ Dictionaries are an unodered collection of zero or more key-value pairs whose keys are object references that refer to hashable objects, and values are object references to object of any type

Example:

```
dict = {'name': "Sujal", "Surname": "vachhani"}
print (dict['name'])
print (dict.get('surname'))
```

Page No.
Date

Output
Sujal
Vachhani

Q.2 Differentiate : Dictionary and List

Ans

Parameter	List	Dictionary
1) Defination	An ordered collection of items	An unordered collection of data in a Key: value pair Form
2) Syntax	Uses square brackets[]	Uses curly braces {}
3) Ordering	Ordered: Items have a defined ordered, which will not change	Unordered: Items do not have a defined order
4) Indexing	Accessed by index, starting from 0	Values are accessed using Keys
5) Mutability	Mutable: Items can be modified after creation	Mutable: Values can be updated, and Key : value pairs can be added or removed
6) Uniqueness	Allow.	
6) Uniqueness	Allows duplicate Items	Does not allow duplicate Keys However, values can be duplicated
7) Example	Fruits = ["apple", "banana"]	person = {"name": "John", "age": 30}
(8) Performance	Faster for ordered operations like sorting	Faster for lookup operations due to the hash mapping of Keys

Q-3 What is the role of Python in Data Scie

Ans Python plays a crucial role in data science due to its simplicity, readability and extensive ecosystem of libraries

(1) Data Cleaning and Manipulation
(2) Data Analysis and Exploration
(3) Data Visualization
(4) Machine Learning and AI
(5) Big Data Processing
(6) Natural Language Processing (NLP)
(7) Automation and Scripting
(8) Community and Open source Libraries

Q-4 Differentiate List and Tuple in Python

Ans

List	Tuple
1) Items are surrounded in square bracket []	1) Items are surrounded in parenthesis()
2) List objects are mutable	2) Tuple objects are Immutable
3) List are slower than tuple	3) Tuples are faster than list
4) Lists consume more memory	4) Tuple consume less memory as compared to the list.
5) List have several built-in methods	5) Tuple does not have many built-in methods
6) Syntax: list1 = [20, 'Ram', 33]	6) Syntax tuple1 = (20, 'Ram', 33)

Q-5 Differentiate : C and Python

Ans

C	Python
→ An imperative programming model is basically followed by C	→ An object-oriented programming model is followed by Python
→ Pointers are available in C language	→ No pointers functionality is available in Python
→ C is compiled language	→ Python is an interpreted language
→ C language is fast	→ Python Programming language is slow.
→ C is complex than Python	→ Python is easier than C
→ There is limited number of built-in functions available in C	→ There is a large library of built-in functions in Python
→ C is statically typed	→ Python is dynamically typed

Q-6 Why Python is very popular language in 21st Century ?

Ans Python has gained immense popularity in 21st Century for several key reasons:

1. Ease of Learning and Readability : Python Syntax is simple and close to natural language, which makes it accessible for beginners
2. Versatility : Python has wide applicability which has attracted developers from various fields
3. Rich Libraries and Frameworks : The Python ecosystem includes extensive libraries and frameworks such as NumPy, Pandas, TensorFlow, Django, Flask and more.

4 Strong Community Support: Python has a large active community that contributes to its development offers support and shares resources, making it easier for new users to learn and solve problems

5 Data Science and AI Boom: Python is the top choice for data science and AI due to its powerful libraries like TensorFlow, Keras and PyTorch

6 Open Source and Community Development: Being open source, Python is freely available and continuously improved by the community. This has encouraged widespread adoption and innovation

o Operators in Python

Operator	Function
+	Addition
-	Subtraction
>>	Right Bit Shift
**	Exponentiation
-	Subtraction
/	Division
<<	Left Bit Shift
%	Modulus

o Features of Python

1) Python is a high-level, interpreted, interactive and object-oriented scripting language
2) It is simple and easy to learn
3) It is portable
4) Python is free and open source programming language
5) Python can perform complex tasks using a few lines of code

6) Python can run equally on different Platforms such as Windows, Linux, UNIX, and Machintosh etc

7) It provides as vast range of libraries For the various fields such as machine learning, web development and also For scripting

Ch-2

Data Science and Python

Q.1 Explain Web Scrapping with Example using Beautiful Soup Library

Ans → Beautiful Soup is a Python library data of HTML and XML files
→ Beautiful Soup helps you all pull particular content from a webpage, remove the HTML markup, and save the information
→ It is a tool for web scraping that helps you clean up and parse the documents you pulled down from the web

- Importing the Beautiful Soup constructor function

```
from bs4 import BeautifulSoup
```

- The BeautifulSoup constructor function takes in two string arguments.
  a. The HTML string to be parsed
  b. Optionally, the name of a parser

Example: Parsing of HTML

```
from bs4 import BeautifulSoup
htmltxt = "<P> Hello World </P>"
Soup = BeautifulSoup(htmltxt, 'xml')
```

- The BeautifulSoup object has a text attribute that returns the plain text of a HTML string

Ch-2

Data Science and Python

Q-1 Explain Web Scrapping with Example using Beautiful Soup Library

Ans → Beautiful Soup is a Python library data out of HTML and XML files

→ Beautiful Soup helps you all pull particular content from a webpage, remove the HTML markup, and save the information

→ It is a tool for web Scraping that helps you clean up and parse the documents you pulled down from the web

- Importing the Beautiful Soup Constructor Function

```
from bs4 import BeautifulSoup
```

- The BeautifulSoup Constructor Function takes in two string arguments.
  a. The HTML string to be parsed.
  b. Optionally, the name of a parser

Example: Parsing of HTML

```
from bs4 import BeautifulSoup
htmltxt = "<P> Hello World </P>"
Soup = BeautifulSoup(htmltxt, 'xml')
```

- The BeautifulSoup object has a text attribute that returns the plain text of a HTML string

In the example of simple soup of <P> Hello World </P> the text attribute returns:
soup.text
'Hello World'

Q-2 Write a brief note on NetworkX library

Ans NetworkX is a Python language software package for the creation, manipulation, and study of the structure, dynamics and functions of complex networks.
→ It is possible to draw small graphs with NetworkX You can export network data and draw with other programs
0 When to Use
1 Unlike many other tools, it is designed to handle data on a scale relevant to modern problems
2 Most of the core algorithms rely on extremely fast legacy code
3 Highly flexible graph implementations

Q-3 List and explain the reasons which make Python programming popular in Data Science

Ans Python is popular in data Science For several reasons which are as Follows:-
1) Rich Libraries: Libraries like NumPy, Pandas, TensorFlow and PyTorch streamline data analysis and machine learning tasks
2) Data Visualization: Tools like Matplotlib and Seaborn allow easy, high-quality visualizations
3) Easy to learn: Python's readable syntax makes it accessible to both beginners and experts, ideal for rapid prototyping

4) Community Support : Python has a strong, active community that provides resources, libraries and support
5) Big Data and Cloud Compatibility: Python integrates with big data tools (eg Spark) and cloud platforms (Aws, GCP, Azure)
6) Versatile and Integrative: Python covers the entire workflow, from data prep to model deployment and integrates well with other languages
7) Open Source: Free and accesible, Python is widely adopted across industries

Q-4 Establish relationship between AI, data science and big data

Ans AI, data science and big data are interconnected fields, each contributing to the other in powerful ways
1) Big data: This field deals with collecting and storing massive datasets from various sources, providing raw material essential for insights and modeling
2) Data Science: Data Science processes and analyzes big data using statistical and computational methods
3) AI and machine learning: AI, especially through machine learning uses data science's insights to build models that automate tasks and make predictions
4) Feedback Loop: AI generates new data through user interaction and outputs which data science reanalyzes to refine insights, continuously improving the entire system

Page No.
Date

Q-5 Provides duties performed by a Data Scientist with suitable example

Ans A data scientist key duties include:-

1) Data Collection and Cleaning: Gathering and preparing data for analysis (eg cleaning transaction records for an e-commerce company)
2) Data Analysis: They analyse data to uncover trends, correlations and outliers Eg: analyzing purchase patterns to target customers.)
3) Feature Engineering: Creating useful data attributes for models (eg developing "payment history" for loan risk models)
4) Building Predictive Models: Developing machine learning models (eg predicting patient readmission risk in healthcare)
5) Data Visualization: Creating visual reports to communicate insights (eg sales dashboards for tracking performance)
6) A/B testing: Running experiments to measure changes (eg testing a new recommendation system for viewer engagement)
7) Collaboration: Working with business teams to align models with goals (eg tailoring credit risk models with risk analysis)

Q-6 Explain Data Science Pipeline in details

Ans → Data Science Pipelines are sequences of processing and analysis steps applied to data for a specific purpose.

1 Preparing the data

→ Big data enables organizations to gather, store manage and manipulate vast amounts of data at the right speed, and the right time, to gain insights

→ The raw data not only may vary substantially in format, but you may also need to transform it to make all the data sources cohesive and amenable to analysis

2. Performing the exploratory data analysis

→ Exploratory Data Analysis (EDA), also known as Data Exploration, is a step in the Data Analysis process, where a number of techniques are used to better understand the dataset being used

→ By conducting EDA, you can turn an almost useable dataset into a completely useable dataset. The use of trial and error is part of the data science art

3 Learning from data: Discovery is part of being a data scientist

4 Visualizing: Visualization means seeing the patterns in the data and then being able to react to those patterns. In other words, data visualizations turn large and small datasets into visuals that are easier for the human brain to understand and process

Q-7 Explain the input function of Python that demonstrates type casting

Ans In Python, the input() function is used to take user input as a string. To convert this input into other data types (type casting), we use functions like int(), float() and str().

Example of type casting with input() in Python

```
age = input("Enter your age:") # takes input as string
age = int(age)  # converting the input to an integer
print("In 5 years, your age will be", age + 5)
```

Output:

```
Enter your age: 6
In 5 years, you will be 11
```

Q-8 Write a Python code to access data from web

Ans

```
import requests
response = requests.get("https://api.exchangerate-api.com/v4/USD")
if response.status_code == 200:
    data = response.json()
    print("Exchange rates for USD:", data['rates'])
else:
    print("Error:", response.status_code)
```

Q-9 Write any four Python command that demonstrate Python as Calculator

Ans Here are four Python command that demonstrate Python's capabilities as a calculator

1) Addition

```
result = 10 + 5
print ("Addition :", result)
```

Output

```
Addition : 15
```

2) Subtraction

```
result = 10 - 5
print ("Subtraction:" result)
```

Output

```
Subtraction : 5
```

3) Multiplication

```
result = 10 * 5
print ("Multiplication :", result)
```

4 Output

```
Multiplication : 50
```

4) Division

```
result = 10/5
print ("Division :", result)
```

Output:

```
Division : 2
```

Q.10 Explain how to create data science pipeline

Ans The following steps are followed to create a data science pipeline

1) Define Objectives: Understand the problem and set goals
2) Data Collection: Gather and load data from sources
3) Data Cleaning: Remove noise, handle missing values, standardize formats
4) EDA: Analyze data patterns and relationships
5) Feature Engineering: Create and select important features
6) Model Training: Train and tune model with relevant algorithm
7) Evaluation: Test model accuracy and performance
8) Deployment: Host model via API's for real time use
9) Monitoring: Track model performance and update as needed
10) Documentation: Document each step for transparency

Q.11 Explain Training and Testing with suitable example

Ans 1) Training Phase
→ In the training phase, the model learns patterns from data. This involves feeding the model with a dataset where both the input features and the corresponding output labels are known

Example: Suppose we are training a model to predict house price. Our dataset might include features like the number of bedrooms, square footage and location, along with the actual house price (label). During training, the model uses these features to learn how different factors impact the price.

2 Testing Phase

In testing phase, we evaluate the model on unseen data to check its generalization ability. The test dataset is seperate from training data, simulating new data model hasn't seen.

Example:

After training a house price model, we test it on a new set of house data to see if it predicts prices accurately. Metrics like Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE) help asses accuracy.

Q-12 What is HTML Parsing?

Ans Check (Q-1)

Page No.
Date

Ch-3
Getting your Hands Dirty with Data

Q.1 Differentiate Numpy and Pandas

Ans

Pandas	Numpy
→ When we have to work on Tabular data, we prefer the Pandas module	When we have to work on numerical data, we prefer the Numpy module
→ The powerful tools of Pandas are DataFrame and Series	Whereas the powerful tool of Numpy is Arrays
→ Pandas consume more memory	Numpy is memory efficient
→ Pandas provides 2d table object called DataFrame	Numpy provides a multi-dimensional array
→ Performance is slower than Numpy	Performance is faster than Pandas
→ It has a higher industry application	It has a lower application
→ Numeric indexing	Numeric indexing only
→ Labeled indexing with row and column names	
→ Usage Example: Data cleaning, analysis and exploration	Scientific computing, machine learning

Q-2 Explain String Slicing in Python with Example

Ans String slicing in Python allows you to extract a substring from a given string by specifying a start and end index. Slicing is done using the syntax

String [start : end : step]

Example

1. Basic Slicing

```
text = "Hello, World!"
print (text[0:5])
```

Output: Hello

2. Using Step

```
text = "Hello, World!"
print (text[0:12:2])  # Output: 'Hlo ol'
```

Output : 'Hlo ol'

→ start is the index where the slice begins (inclusive)
→ end is the index where the slice stops (exclusive)
→ step is the interval at which characters are selected (optional)

Q.3 Differentiate rand and randn function in Numpy

Ans

Feature	rand	randn
1) Purpose	Generates random samples from a uniform distribution	Generates random samples from a standard normal (Gaussian) distribution
2) Distribution	Uniform distribution between [0, 1)	Standard normal distribution (mean = 0, std deviation = 1)
3) Range of Values	Values are between 0 and 1	Values can range from $-\infty$ to $+\infty$
4) Typical output shape	rand(d0, d1, .. dn) for desired shape	randn (d0, d1, .. dn) for desired shape
5) Example Output	array [0.72, 0.15, 0.99]	array [-1.01, 0.87, 0.32]
6) Use Case	Ideal for generating random proportions, probabilities or weights	Ideal for simulations, statistics, or models requiring Gaussian noise

Q.4 Explain DataFrame in Pandas with Example

Ans A DataFrame in Pandas is a two-dimensional, labeled data structure similar to table in a relational database or an Excel spreadsheet. It consists of rows and columns and can hold data of different types (such as integers, float, strings etc) in each column.

Example

```
import Pandas as pd

data = {
    'Name' : ['Ram', 'Shyam', 'Raju'],
    'Age'  : [21, 22, 23]
    'City' : ['Surat', 'Delhi', 'Mumbai']
}

df = pd.DataFrame(data)
print(df)
```

Output

	Name	Age	City
0	Ram	21	Surat
1	Shyam	22	Delhi
2	Raju	23	Mumbai

Q-5 Explain Groupby Function in Pandas with example

Ans The Groupby function in Pandas is used to split the data into groups based on some criteria, apply to function to each group and then combine the results back into single DataFrame. It is commonly used for aggregation, transformation, and filtering of data

```
import pandas as pd

data = {
    'Name' : [ 'Ram', 'Rahul', 'Arjun', 'Varun', 'Roy' ],
    'Subject' : [ 'Math', 'Math', 'Science', 'Math', 'Science' ],
    'Score' : [ 85, 90, 95, 80, 88 ]
}

df = pd.DataFrame(data)

# Group by 'Subject' and calculate average 'Score'
grouped = df.groupby('Subject')['Score'].mean()
print(grouped)
```

Output:

```
Math      85.0
Science   91.5
Name: Score, dtype: float64
```

Q-5 Explain how to deal with missing data in Pandas

Ans In Pandas, handling missing data is a crucial part of data cleaning and preparation. Missing data can appear in various form such as NaN (Not a Number) or None

1) Detecting Missing Data
To identify missing values, Pandas provides several methods
- isnull(): Returns a DataFrame of the same shape but with True for missing values and False for non-missing
- notnull(): Opposite of isnull(), it returns True for non-missing values

```
import pandas as pd
import numpy as np

# Sample data with missing values
data = {
    'Name': ['Ram', 'Shyam', 'Raj', 'Ray', np.nan],
    'Age': [24, np.nan, 23, 25, 29]
}

df = pd.DataFrame(data)

# Detect missing values
print(df.isnull())
```

Output:

	Name	Age
0	False	False
1	False	True
2	False	False
3	False	False
4	True	False

Q-7 Differentiate Join and Merge Functions in Pandas

Ans

	Feature	Join	Merge
1)	Purpose	Combines two DataFrames based on the index or a column	Combines two DFs based on one or more column or indices
2)	Default Key	Uses the index by default	Uses columns

3	Join Types	Supports left, right, outer and inner joins	Supports left, right, outer and inner joins
4	Result	Returns a DataFrame with an index-based join	Returns a DataFrame with the specified join columns
5	Usage	Best used for joining on index or simpler operations	Best used for more complex joins, especially on columns
6	Syntax Simplicity	Simpler for joining on index	More complex than Join

Q-8 Explain Bag of Word Model [03m] OR Elaborate a bag of word concept in detail [07m]

Ans The Bag of worlds model is a method of text representation where each document (sentence, paragraph or article) is represented as a collection of words - a "bag"

Each document in a dataset is transformed into a fixed-length vector that represents the counts of each word in the vocabulary across all documents

Example
Suppose we have two short sentences
1 I love dogs
2 I love cats

Step 1 Build vocabulary
→ Vocabulary : ["I", "love", "dogs", "cats"]

Step 2: Create Vectors for Each Sentence

Word	Sentence 1("I love dogs")	Sentence 2("I love cats")
I	1	1
love	1	1
dogs	1	0
cats	0	1

Step 3: Represent Each Sentence as a vector

Sentence 1: [1, 1, 1, 0] (1 "I", 1 "love", 1 "dogs", 0 "cats")

Sentence 2: [1, 1, 0, 1] (1 "I", 1 "love", 0 "dogs", 1 "cats")

Final Bag of Words Representation

	I	love	dogs	cats
Sentence 1	1	1	1	0
Sentence 2	1	1	0	1

We can now use to these vectors in machine learning models to find similarities or make predictions based on word presence and frequency

Q-9 Explain following string functions with suitable example
len, count, title, lower, upper, find, rfind, replace

Ans 1) len
→ Purpose: Returns the length (number of characters) of a string

→ Example
text = "Hello World"
print (len(text))

Output: 11

2) Count()
→ Purpose: Counts the occurences of a specific substring in a string
→ Example
text = "banana"
print (text.count("a"))

Output: 3

3) title()
→ Purpose: Converts the first letter of each word in the string to uppercase
→ Example
text = "hello"
print (text.title())

Output: Hello

4) lower()
Purpose: Converts all characters in the string to lowercase
→ Example
text = "Hello"
print (text.lower())

Output: hello

5. Upper()
→ Purpose: Converts all characters in the string to uppercase
→ Example
```
text = "hello world"
print (text.upper())
```

Output: Hell. HELLO WORLD

6. Find()
→ Purpose: Returns the index of first occurence of a substring in the string. Returns -1 if the substring is not found

Example
```
text = "hello world"
print (text.find ("world"))
print (text.find ("python"))
```

Output:
6
-1

7. rfind()
→ Purpose: Returns the index of the last occurence of a substring. Returns -1 if the substring is not found
Example
```
text = "banana"
print (text.rfind ("a"))
```
Output: 5

8 replace()
→ Purpose: Replaces occurences of a specified substring with another substring

Example
```
text = "hello world"
print (text.replace ("world", "Python"))
```

Output
hello Python

Q-10 Explain stemming in detail with relatable example

Ans Stemming is a natural language processing (NLP) technique used to reduce words to their base or root form
→ Stemming allows for easier text processing, comparision and analysis in applications like search engines, information retrieval and machine learning
→ Stemming algorithms remove suffixes or prefixes from words to produce a base form.
→ The result may not always be a "real" word, but it serves the purpose of grouping related words together
Example
→ Consider the words: "running", "runs" and "runner"
→ A stemming algorithm might reduce all of these words to the common root "run"
→ This helps in treating all variations of "run" as a single concept

Example Code Using NLTK's Porter Stemmer

```
From nltk.stem import PorterStemmer

Stemmer = PorterStemmer()

Words = ["running", "runner", "ran", "runs", "easily", "fairness"]

Stems = [Stemmer.stem(word) for word in words]
print (stems)
```

Output:

['run', 'runner', 'ran', 'run', 'easili', 'fair']

Q-11 Elaborate XPath in detail with relatable example

Ans XPath is a query language used for selecting nodes from an XML document. It is commonly used for data extraction in web scraping and working with XML data structures, such as HTML documents in web pages

```
<html>
  <body>
    <div class: "container">
      <h1> Welcome to the store </h1>
      <div class: "product" id: "p1">
      <span class: "name"> Laptop </span>
      <span class: "price"> $1200 </span>
      </div>
```

```
<div class="product" id="h2">
<span class="name">Smartphone</span>
<span class="price">$800</span>
</div>
</body>
</html>
```

1) Extract all product names
`// div[@class="product"]/span[@class="name"]/text()`

Output : ["Laptop", "Smartphone"]

2) Extract all product prices
`// div[@class="product"]/span[@class="price"]/text()`

Output : ["$1200", "$800"]

Q-12 List and Explain any three Magic Function

Ans
1. __init__ method
2. __str__ method
3. __len__ method

1 __init__()
→ Purpose: Initializes an object's state when it is created
Often refferred to as Constructor
→ Explanation: When you create an instance of a class, Python calls the __init__() method to set up the object's initial attributes

2. __str__()

→ Purpose: Defines the string representation of an object used when print() or str() is called on the object

→ Explanation: The __str__() method allows you to customize how an object is represented as a string, making output more readable

3. __len__()

→ Purpose: Defines the behaviour of the len() function when it's called on an object.

→ Explanation: The __len()__() method allows you to specify how an object should be measured for length, making it compatible with the len() function.

Q-13 Explain Slicing rows and columns with Example

Ans In Python, Slicing rows and columns refers to selecting specific parts of a dataset, typically using Pandas, a library that makes data manipulation easy

```
import Pandas as Pd

data = {
    'Name': ['Alice', 'Ram', 'Shyam', 'Ravi', 'Eva'],
    'Age': [24, 25, 26, 27, 28],
    'City': ['Surat', 'Rajkot', 'Delhi', 'Mumbai', 'NYC']
}

df = Pd.DataFrame(data)
print(df)
```

Page No.
Date

- Slicing Rows

Slicing rows by index Range:

→ You can select specific rows by their index position using the iloc method

Example: Select the first three rows

```
print (df.iloc [0:3])
```

	Name	Age	City
0	Alice	24	Surat
1	Ram	25	Rajkot
2	Shyam	26	Delhi

- Slicing Columns

→ For slicing columns by name you can specify columns by their names

```
print (df [['Name', 'Age']])
```

Output

	Name	Age
0	Alice	24
1	Ram	25
2	Shyam	26
3	Ravi	27
4	Eva	28

Q-14 What do you mean by missing values ? Explain different ways to handle the missing values with example

Ans Missing values in a dataset refer to the absence of data in some fields
they occur when no value is stored for a variable in an observation often represented as NaN (Not a Number) in Python's Pandas library

1) Removing Missing Values

```
import Pandas as Pd
import numpy as np

data = {
        'Name': ['Alice', 'Bob', np.nan, 'Ram', 'Sujal'],
        'Age': ['24', np.nan, '22', '23', '20'],
        'City': [np.nan, 'Surat', 'Rajkot', 'Delhi', 'NY
    }
df = Pd.DataFrame(data)

# Drop rows with missing values
df_cleaned = df.dropna()
print(df_cleaned)
```

Output

	Name	Age	City
3	Ram	23	Delhi
4	Sujal	20	NYC

2) Forward Fill and backward Fill : Filling missing values with the previous (ffill) or next (bfill) value in the column

```
# Forward Fill missing values in the "Name" Column
df['Name'].fillna(method='ffill', inplace=True)
print(df)
```

Output

	Name	Age	City
0	Alice	24	~~nan~~ NaN
1	Bob	NaN	Surat
2	Bob	22	Rajkot
3	Ram	23	Delhi
4	Sujal	20	~~Mumb~~ NYC

3) Imputing Missing Values

o Mean/Median/Mode Imputation : Replacing missing values with the mean, median or mode of the column

```
# Impute missing age with Mean
df['Age'].fillna(df['Age'].mean(), inplace=True)
print(df)
```

	Name	Age	City
0	Alice	24	~~nan~~ NaN
1	Bob	22.5	Surat
2	NaN	22	Rajkot
3	Ram	23	Delhi
4	Sujal	20	NYC

Q-15 What is use of following operations on Pandas Data Frames ?
Explain with a small example of each
1. shape 2. tail() 3. describe()

Ans 1) shape :

Purpose: The shape attribute returns the dimensions of the DataFrame as a tuple (number of raws, number of columns)
Example
import pandas as pd

```
data = {
        'Name' : ['Ram', 'Shyam', 'Charlie', 'David', 'Eva'],
        'Age' : [24, 27, 22, 32, 29]
        'City' : ['NYC', 'LA', 'Chicago', 'Houston', 'Phoenix']
       }
df = pd.DataFrame(data)

print (df.shape)
```

Output
(5, 3)

2) tail()
→ Purpose: The tail() method returns the last few rows of the DataFrame. By Default, it shows the last 5 rows but you can specify a different number

```
print (df.tail(2))
  # Shows the last 3 rows of DataFrame
```

Output

	Name	Age	City
2	Charlie	22	Chicago
3	David	32	Houston
4	Eva	29	Phoenix

3) Describe()

→ Purpose: The describe() method provides summary statistics for numerical columns, including count, mean, standard deviation, min, max and percentiles.

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

Output

	Age
count	5.0000
mean	26.8000
std	3.62323
min	22.0000
25%	25.0000
50%	27.0000
75%	29.0000
max	32.0000

Q.16 What are the different ways to remove duplicate values from dataset?

Ans Removing duplicate values from a dataset is essential for accurate analysis, as duplicates can distort insights and predictions.

1. Remove Duplicate Rows Using drop_duplicates()

→ Basic Usage: The drop_duplicates() method removes duplicate rows. By default, it keeps the first occurence and removes later duplicates

```
import pandas as pd

data = {
    'Name': ['Alice', 'Bob', 'Alice', 'David', 'Bob'],
    'Age': [24, 27, 24, 32, 27]
    'City': ['New York', 'Los Angeles', 'New York', 'Houston', 'Los An
}

df = pd.DataFrame(data)

df_no_duplicates = df.drop_duplicates()
print(df_no_duplicates)
```

Output

	Name	Age	City
0	Alice	24	New York
1	Bob	27	Los Angeles
3	David	32	Houston

2. Remove Duplicates Based on Specific Columns

```
df_unique_names = df.drop_duplicates(subset=['Name'])
print(df_unique_names)
```

Output

	Name	Age	City
0	Alice	24	New York
1	Bob	27	Los Angeles
3	David	32	Houston

3 Identifying and Removing Duplicate Rows with duplicated()

```
# Find and display duplicate rows
duplicates = df [df.duplicated()]
print (duplicates)
```

Output

	Name	Age	City
2	Alice	24	New York
4	Bob	27	Los Angeles

```
# Remove duplicates based on 'Name' and 'City'
df_no_duplicates = df [~df.duplicated()]
```

Q-17 Is String a mutable data type? Also explain the string operations length, slicing and indexing in detail with an appropriate example

Ans In Python, strings are immutable data types, meaning once a string is created, it cannot be modified

0 String Operations
1) Length (len)
→ the len() function returns the total number of characters in string

Example

```
S = "Hello world"
print (len(s))
```

Output: 11

8) Indexing

→ Indexing allows us to access individual characters in a string by their position. Indexing starts from 0 for the first character and goes upto to len(s) - 1

Example

```
S = "hello"
print (s[0])
print (s[4])
print (s[-1])
print (s[-2])
```

Output

```
h
o
o
l
```

9) Slicing

→ Slicing is used to extract a part (substring) of a string. The syntax is s[start:end] where

- start is the index where slicing begins (inclusive)
- end is the index where slicing stops (exclusive)

Q-18

Ans

Page No.
Date

Example
```
S = "hello world"
print (S[0:5])
print (S[6:0])
print (S[:5])
print (S[-5:])
```

Output
```
hello
world
hello
world
```

Q-18 Explain %matplotlib magic function

Ans The %matplotlib magic function in Python is used to configure the matplotlib plotting library to display plots inline or in an external window when running code in an IPython environment

Syntax:
```
%matplotlib [backend]
```

Commonly Used Backend are
1) inline
2) notebook
3) qt
4) agg

Q.19 What are the magic functions in Jupyter? Explain with example

Q.20

Ans

Ans 1) %time
→ Measures the execution time of single line of code

Eg:
%time sum([i for i in range(1000000)])

2) matplotlib
Ensures that matplotlib plots are displayed inline
Eg:

%matplotlib inline
import matplotlib.pyplot as plt
plt.plot([1,2,3,4], [1,4,9,16])
plt.show()

(3) %run
Runs a Python script as if it were a standalone program
Eg:
%run my_script.py

(4) %load
Loads code from a file or URL and inserts it into the notebook
Eg: %load my_script.py

2)

Page No.
Date

8-20 Write a small code to perform following operations on data: Slicing, dicing, Concatenation, Transformation

Ans 1) Slicing
→ It refers to selecting specific rows or columns from a Data frame

```
import pandas as pd

data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David'],
    'Age': [23, 34, 45, 29]
    'City': ['New York', 'Los Angeles', 'Chicago', 'Houston']
}

df = pd.DataFrame(data)
sliced_df = df[['Name', 'City']]
print(sliced_df)
```

Output:

	Name	City
0	Alice	New York
1	Bob	Los Angeles
2	Charlie	Chicago
3	David	Houston

2) Dicing
Dicing involves filtering data based on specific conditions. Here we'll filter rows where Age is greater than 30

Filter rows based on a condition
```
diced_df = df[df['Age'] > 30]
print(diced_df)
```

	Name	Age	City
1	Bob	34	Los Angeles
2	Charlie	45	Chicago

3. Concatenation
Concatenation is used to combine two or more DataFrames. For simplicity, we'll concatenate df with itself, creating duplicate rows

```
concatenated_df = pd.concat([df, df], ignore_index = True)
print(concatenated_df)
```

Output:

	Name	Age	City
0	Alice	23	New York
1	Bob	34	Los Angeles
2	Charlie	45	Chicago
3	David	29	Houston
4	Alice	23	New York
5	Bob	34	Los Angeles
6	Charlie	45	Chicago
7	David	29	Houston

4 Transformation
Transformation applies a function to modify data. Here we'll increase each person's Age by 5

transformed_df = df

transformed df

```
df['Age'] = df['Age'] + 5
print(df)
```

Output

	Name	Age	City
0	Alice	28	New York
1	Bob	39	Los Angeles
2	Charlie	50	Chicago
3	David	54	Houston

Q-21 Describe the three levels of Flat-File dataset. Explain any two with the help of example [07m]

Ans 1) Simple Flat File
2) Hierarchial Flat File
3) Relational Flat File

1) Simple Flat File
A simple flat file consists of single, two-dimensional table where data is stored in rows and columns. Each row represents a unique record, and each column represents a specific attribute or field related to that record
→ This type of file doesn't support relationships or dependencies between records

Example

Consider a CSV file named Students.csv with following content

Student ID	Name	Age	Grade
1	Alice	20	A
2	Bob	21	B
3	Charlie	22	A

2) Hierarchial Flat Files

A Hierarchial Flat File organizes data in parent-child structure, where a single primary table associated with one or more secondary tables

→ This type of dataset structure is typically used when records in one table depend on records in another. Consider a file with two tables : employees.csv (primary table) and projects.csv (secondary table)

employees.csv

Employee ID	Name	Department
1	John	Marketing
2	Sarah	IT

projects.csv

Project ID	Employee ID	Project Name
101	1	Market Research
102	1	Brand Strategy
103	2	Software Upgrade
104	2	Cybersecurity

Ch-4 Data Visualization

Q-1 List and Explain different graphs in MatPlotLib ?

Ans The Following types of graphs are present in MatPlotlib are:-

1) Pie Chart
2) Bar Chart
3) Histograms
4) Box Plots
5) Scatter Plot

1) Pie Chart

→ A Pie chart is a circular chart divided into slices, where each slice represents a proportion of the total

→ Pie charts are ideal for showing the composition of a dataset in terms of categorical proportions

Example

```
import matplotlib.pyplot as plt

labels = ['A', 'B', 'C']
Sizes = [30, 40, 30]

plt.pie( sizes, labels = labels, autopct = '%1.1f%%')
plt.show()
```

2) Bar Chart

→ A bar chart represents data with rectangular bars where the length of each bar corresponds to the value of the category. Bars can be displayed vertically or horizontally, allowing for an easy comparision of quantities

Code:

```
import matplotlib.pyplot as plt

categories = ['A', 'B', 'C']
values = [5, 7, 3]

plt.bar(categories, values)
plt.show()
```

3) Histogram

→ A histogram is a graphical representations of the distribution of numerical data, dividing the data into intervals called bins. Each bin's height indicates the frequency of data points within that interval, making histograms valuable for understanding data distribution

Code

```
import matplotlib.pyplot as plt

data = [1, 2, 2, 3, 3, 4, 4, 5, 5, 5, 6]

plt.hist(data, bins = 5)
plt.show()
```

4) Box Plot

→ A boxplot also known as box and whisker plot, summarizes a dataset by displaying the median, quartiles and potential outliers.

→ Boxplots are ideal for comparing distributions across multiple datasets, such as test scores by class, heights by age group or sales by region

Example Code

```
import numpy as np
data = np.random.normal(0, 1, 100)
plt.boxplot(data)
plt.ylabel('Value')
plt.title('Boxplot of sample data')
plt.show
```

(5) Scatter Plot

→ A scatter plot shows individual data points on a two dimensional grid, where each point represents a pair of values from two variables. It's used to reveal the relationship if any, between the two variables

→ Scatter plots are commonly used in correlation analysis. such as investigating the relationship between weight and weight, study time and exam score, or marketing spend and revenue

Example Code

```
x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 10]
plt.scatter(x, y, color = 'blue')
plt.xlabel('X-axis')
```

plt.ylabel('Y-axis')
plt.title('Scatter Plot example')
plt.show

6) Time Series Plot
A time series plot is used to display data points at successive time intervals, allowing us to visualize trends over time

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

dates = pd.date_range(start="2023-01-01", periods=10)
values = np.random.randint(1, 10, size=10)

plt.plot(dates, values)
plt.xlabel("Date")
plt.ylabel("Value")
plt.title("Time Series Plot")
plt.xticks(rotation=45)
plt.show()
```

7) Geographical Data Plot
→ Geographical data plotting allows you to visualize data points on a map, often used to show location-based data like temperature, sales, regions or geographic distributions

Q-2 Explain Labels, Annotation and Legends in Matplotlib

Ans 1) Label : Make it easy for the viewer to know the name or kind of data illustrated
2) Annotation : Help extend the viewer's knowledge of the data, rather than simply identify it
3) Legend : Provides cues to make identification of the data graph easier.

Q-3 Elaborate Graphs along with its types [07m]

Ans See Q-1

Q-4 Differentiate Bar graph Vs Histogram

Ans

	Feature	Bar Graph	Histogram
1)	Defination	Display discrete categories or graphs	Displays continuous data divided into intervals
2)	Dimensionality	One Dimension	Two-Dimension
3)	Representation of Frequency	Length of the bars	Area of the bar
4)	Spacing between bars	Bars are seperated with equal spaces	Bars are touching each other
5)	Purpose	Compares different categories	Shows frequency distribution of continuous data
6)	Example	Sales by product type, Population by Country	Age distribution, exam scores, height distribution

Page No.
Date

Q-4 Provide explanations on the importance of Grafhs in Data Science

Ans The importance of Grafhs in Data Science is due to following

(1) Simplify Data : Grafhs make big, complex data easier to understand at a glance

(2) Spot Patterns: They help us see trends, patterns and changes over time

(3) Easy Comparision : Grafhs make it simple to compare different groups or categories

(4) Better decisions: Clear visual help teams make quick, informed decisions based on the data

(5) Find Outliers: They help identify unusual data points, which may indicate errors or unique cases

(6) Improve Communication: Grafhs make data more engaging and easier for everyone to understand

(7) Explore Data : Visualizations let data scientists interact with data to find insights

(8) Show Relationships : Charts like Scatter plots reveal how two variables may be connected

Q-5 Depict steps to create a Scatter plot with example

Ans Steps to create a Scatter Plot

1 Import Libraries : Import the necessary libraries, usually matplotlib.pyplot for plotting

2 Prepare data : Have two lists or arrays representing the x and y data points

3 Create Plot: Use plt.scatter() to create scatter plot

4 Add labels and Title: Label your axes and add title for clarity

5. Show Plot: Use plt.show() to display the plot

Code:

```
import matplotlib.pyplot as plt

study_hours = [1, 2, 3, 4, 5, 6, 7, 8]
test_scores = [60, 55, 60, 62, 70, 75, 78, 85]

plt.scatter(study_hours, test_scores, color='blue', marker='o')

plt.xlabel("Study hours")
plt.ylabel("Test Scores")
plt.title("Study Hours Vs Test Scores")

plt.show()
```

Q-6 Why data visualization is important in Data Science?

Ans Data Visualization is important in Data Science due to following reasons :-

1) Simplifies Complex Data
2) Reveals Insights Quickly
3) Enables Pattern and trends detection
4) Improves Communication
5) Enhances Decision Making
6) Supports Comparitive Analysis

Q.7 Explain bar() function with code

Ans The bar() function in Matplotlib is used to create a bar chart, where each bar represents a category or data point, and the height represents a value

Syntax
matplotlib.pyplot.bar(x, height, width, color) (optional)

Code
import matplotlib.pyplot as plt

x = ['A', 'B', 'C', 'D']
y = [3, 7, 1, 5]

plt.bar(x, y)

plt.xlabel("Categories")
plt.ylabel("Values")
plt.title("Bar Chart")

plt.show

Q.8 Explain hist() function with code

Ans The hist() function in Matplotlib is used to create a histogram, which is a graphical representation of data distribution. A histogram groups data into bins and shows the frequency of data points within each bin

Syntax
matplotlib.pyplot.hist(x, bins, color...)

```
import matplotlib.pyplot as plt

ages = [21, 22, 23, 24, 25, 26]

plt.hist(ages, bins=5, color='blue')

plt.xlabel("Ages")
plt.ylabel("Frequency")
plt.title("Age Distribution")

plt.show()
```

Q-9 Explain box plot with example

Ans See Q-1

Q-10 Explain scatterPlots with example

Ans See Q-1

Q-11 Write a code to draw Pie chart using Python's Library

Ans See Q-1

Q-12 What do you understand by Data Visualization? Discuss Some Python's data visualisation techniques

Ans

Page No.
Date

Data Visualization is the process of converting raw data into graphical or visual formats such as charts, graphs, maps and plots to help easily understand trends, patterns, outliers and insights in the data

Data Visualization Techniques

1) Matplotlib
   → Line Plot
   → Bar Chart
   → Histogram
   → Scatter Plot
   → Pie Chart

2) Seaborn
   → HeatMap
   → Box Plot
   → Violin Plot
   → Pair Plot

3) Plotly
   → Interactive
   → 3D plots
   → Geo Maps

4) Pandas Visualization
   → Line Plots
   → Bar and Histogram Plots
   → Box Plots

Q-13 List the different line styles used in charts

Ans The different line styles used in charts are

1) Solid Line
→ Line Style Code: '-'
→ Example : ——

2) Dashed Line
→ Line style code: '--'
→ Example : - - - -

3) Dash-dot line
→ Line style code: '-.'
→ Example -.-.-.

4) Dotted line
→ Line style code: ':'
→ Example: ....

5) No line
→ Line style code: ' ' or None
→ Example:

Q-14 Explain with the example use of axes() and grid() method

Ans 1) axes()
The axes() method in Matplotlib allows you to create a new set of axes in the current figure or to customize the limits and appearance of an existing plot

Example of axes() to Set axes limits

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [10, 15, 13, 18, 20]

plt.plot(x, y, marker='o')

plt.axes().set_xlim(0, 6)
plt.axes().set_ylim(5, 25)

plt.xlabel("X axis")
plt.ylabel("Y axis")
plt.title("Plot with Custom Axes limits")

plt.show()
```

2) grid()

The grid() function in Matplotlib enables or disables grid lines in a plot, which can improve readability by making it easier to compare values across axes

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [10, 15, 13, 18, 20]

plt.plot(x, y, marker='o')

plt.grid(True, linestyle='--', color='gray', linewidth=0.7)
```

```
plt.xlabel("X axis")
plt.ylabel("Y axis")
plt.title("Plot with Grid lines")

plt.show()
```

Q-15 Write a Python code for to draw two lines on plot. Both lines should be depicted with different colors

Ans

```
import matplotlib.pyplot as plt

x1 = [1, 2, 3, 4, 5]
y1 = [2, 3, 5, 7, 9]  # Line 1 data

x2 = [2, 3, 6, 7, 8]
y2 = [4, 6, 7, 9, 10]  # Line 2 data

plt.plot(x1, y1, color='blue', label='line 1')

plt.plot(x2, y2, color='red', label='line 2')

plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Two lines with diff. colours")

plt.legend()

plt.show()
```

Q-16 What is the use of labels and legends in plot? Explain with example

Ans (1) Labels
→ Labels are used to describe the x-axis and y-axis of a plot, giving context to what each axis represents.

xlabel(): Adds a label to the x-axis
ylabel(): Adds a label to y-axis

2) Legends
→ A legend is used to identify different data series or elements within a plot, especially when there are multiple lines, bars or markers.

legend(): Displays the legend on the plot

Example
```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y1 = [1, 4, 9, 16, 25]
y2 = [2, 3, 5, 7, 11]

plt.plot(x, y1, color='blue', label='line 1')

plt.plot(x, y2, color='red', label='line 2')

plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Two lines with diff. colors")
plt.legend()
plt.show()
```

Q.10 Write a Python code to draw a bar chart adding at least three properties of it

Ans

```
import matplotlib.pyplot as plt

categories = ['Category A', 'Category B', 'Category C', 'Category D']
values = [10, 15, 17, 2]

plt.bar(categories, values,
        color='skyblue'   # P1
        width=0.5         # P2
        edgecolor='black' # P3
       )

plt.xlabel("Categories")
plt.ylabel("Values")
plt.title("Bar Chart with Custom Properties")

plt.show()
```

Q.11 Explain Importance of Legends, Labels and Annotations in Graphs

Ans

1) Legends
   - Importance
   1) Identification
   2) Ease of Interpretation
   3) Clarity

2) Labels
   - Importance

1) Context
2) Accuracy
3) Professional Presentation

3) Annotations
e) Importance
1) Highlight Key Data Points
2) Add Explanations
3) Enhance Storytelling

Ch-5 Data Wrangling

Q-1 Explain Exploratory Data Analysis (EDA)

Ans Exploratory Data Analysis (EDA) refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations

→ EDA is an approach/philosophy for data analysis that employs a variety of techniques to

1) Maximize insight into a data set
2) Uncover underlying structure
3) Extract important variables
4) Detect outliers and anomalies
5) Develop parsimonious models
6) Test underlying assumptions
7) Determine optimal factor settings

Q-2 Explain Hashing trick in Python with example

Ans Most Machine Learning algorithms uses numeric inputs, if our data contains text instead we need to convert those text into numeric values first, this can be done by using hashing trick

Eg:

```
from sklearn.feature_extraction.text import HashingVectorizer

texts = ["cat dog", "dog fish", "cat fish"]
```

vectorizer = HashingVectorizer(n_features=4)

hashed_features = vectorizer.transform(texts)

print(hashed_features.toarray())

Q3) Differentiate Supervised and Unsupervised Learning

Ans

Supervised Learning	Unsupervised Learning
1) Supervised Learning algorithms are trained using labeled data	1) Unsupervised learning algorithms are trained using unlabeled data
2) Supervised learning model takes direct feedback to check if it is predicting correct output or not	2) It does not take any feedback
3) The number of classes is known	3) Is not known
4) The desired output is given	4) The desired output is not given
5) Supervised model produces an accurate result	5) Model may give less accurate result as compared to supervised learning
6) In this model training data is used to infer model	6) Here, training data is not used
7) Eg: Optical Character Recognition	7) Eg: Find a face in an image

Q-1 Explain Regression with example

Ans Regression is a type of supervised machine learning technique used to predict a continuous target variable based on one or more input features.

→ It aims to find the relationship between the independent variables (input) and the dependent variable (output)

Example

Imagine we want to predict a house price based on its size

Size (sq ft)	Price (in $1000)
500	150
1000	300
1500	450
2000	600

Python Code

```
from sklearn.linear_model import LinearRegression
import numpy as np

X = np.array([500, 1000, 1500, 2000]).reshape(-1,1)  # Size
Y = np.array([150, 300, 450, 600])                   # Price

model = LinearRegression()
model.fit(X, Y)

predicted_price = model.predict([[1200]])  # predict price for 1200 SQFT

print(f"Predicted price for a 1200 sq ft house: ${predicted_price}")
```

Output:
Predicted Price for a 1200 sq ft house: $360

Q.5 Explain Classification with Example

Ans Classification is a type of supervised machine learning technique used to predict a discrete class label for given input data
→ It involves categorizing data into predefined groups based on its features
Eg: Spam detection, disease diagnosis and image recognition

Example
Classify a fruit as either Apple or Orange based on its weight and texture

Data:

Weight (grams)	Texture (0 = Smooth, 1 = bumpy)	Label
150	0	Apple
170	0	Apple
140	1	Orange
130	1	Orange

Example:
```
from sklearn.tree import DecisionTreeClassifier

X = [[150,0], [170,0], [140,1], [130,1]]
y = ["Apple", "Apple", "Orange", "Orange"]

model = DecisionTreeClassifier
model.fit(X,y)
```

```
new_fruit [[160, 0]]
prediction = model.predict(new_fruit)

print(f"The new fruit is: {prediction[0]}")
```

Output:
The new fruit is: Apple

Q-6 Define Covariance and Correlation

Ans Covariance is a measure of how two random variables change together. It quantifies the degree to which changes in one variable are associated with changes in other

Correlation measures the strength and direction of the linear relationship between two variables. It standardizes Covariance to make it dimensionless and easier to interpret

Q-7 Define Correlation and explain its importance in Data Science

Ans Correlation measures the strength and direction of the linear relationship between two variables. It standarizes covariance to make it dimensionless and easier to important

0 Importance of Correlation in Data Science
1) Feature Selection
→ Identifying relationships between variables helps in selecting relevant features for machine learning models
2) Understanding Data Relationships
→ Correlation provides a quick way to understand how

variables influence one another

3) Hypothesis Testing
→ Correlation can be used to test hypotheses, such as "Does increasing temperature correlate with higher ice cream sales?"

4) Data Visualization
→ Visualizing correlation using scatter plots or heatmaps provides a clearer picture of data relationships

5) Improving Model Performance
→ Highly correlated features can lead to redundant information. Identifying and removing them can reduce overfitting and improve computational efficiency

6) Risk Analysis
→ In financial data science, correlation between assets (eg. stocks) is crucial for portfolio diversification and risk assessment.

Q-8 Define Covariance and explain its importance with appropriate example

Ans Importance
1) Identifying Relationships between variables
2) Feature selection in Machine Learning
3) Portfolio Management
4) Understanding Data Structure
5) Model Performance Evaluation
6)

Example:
Scenario: Sales and Advertising Spend
Let's say we have data on advertising spend and sales revenue for a company. We want to know if there is a relationship between two variables

Advertising Spend ($)	Sales Revenue
1000	2000
1500	3000
2000	4000
2500	5000

1) Step 1: Calculate the mean of both variables
2) Step 2: Compute the variance
3) Step 3: Interpretation

→ A positive covariance suggests that the two variables are positively related. As the company spends more on advertising the sales revenue tends to increase

Q-9 What is Scikit-Learn?

Ans Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests and K-nughbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy

→ The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes

1) NumPy: Base n-dimensional array package
2) SciPy: Fundamental library for scientific computing
3) Matplotlib: Comprehensive 2D/3D plotting
4) IPython: Enhanced interactive console
5) Sympy: Symbolic mathematics
6) Pandas: Data Structures and Analysis

Q-10 What is Categorical Variables? Explain it with example

Ans Categorical variables are variables that represent distinct groups or categories. Unlike numerical variables, they do not have a natural order or a quantative meaning. They are usually used to describe attributes or characteristics

Type of Categorical Variables

1) Nominal : Categories have no natural order
→ Example : Colors of cars (Red, Blue, Green)
2) Ordinal : Categories have a logical order
→ Example : T-shirts (Small, Medium, Large)

Example

1) Gender (Nominal)
→ Mall, Female, Other
→ These are categories with no inherent ranking

2) Education Level (Ordinal)
→ High school, UG, PG
→ These categories have an order of progression

3) Car Brands (Nominal)
→ Toyota, Honda, Ford
→ No natural order exists between brands

4) Survey Ratings (Ordinal)
→ Very Dissatisfied, Dissatisfied, Neutral, Satisfied, Very Satisfied
→ These have a ranking with "Very Satisfied" being the highest

Q-1 What is Data Wrangling process? Define Exploratory Data Analysis? Why EDA is required in data analysis?

Ans Data wrangling is the process of transforming data from its original "raw" form into a more digestible format and organizing sets from various sources into a singular coherent whole for further processing

o Data wrangling covers the following processes

1) Getting data from the various source into one place
2) Piecing the data together according to the determined setting
3) Cleaning the data from the noise or erroneous, missing elements

There are six iterative steps in data wrangling process:-

1) Discovering
2) Structuring
3) Cleaning
4) Enriching
5) Validating
6) Publishing

EDA: Chuck Q-1

EDA is an essential step in any data analysis project for the following reasons

1) Understanding the data
→ Helps analysts grasp the data structure, distribution and key trends
2) Identifying Data Quality Issues
→ Highlights missing values, outliers and inconsistencies that need correction

3) Generating Hypotheses
→ Aids in forming hypotheses for further statistical or predictive modeling
4) Feature Selection
→ Identifies important variables or features relevant to the problem at hand
5) Model Preparation
→ Ensures the data is clean and structured, enabling better performance of machine learning models
6) Risk mitigation
→ Helps avoid incorrect conclusions by addressing issues such as skewed data or irrevelant variables

Q-12 Explain Z-Score Standarization OR Define Standarization. Explain Z-Score Standarization with suitable example

Ans Standarization is a data preprocessing technique used to scale features so that they have a mean of 0 and a standard deviation of 1
→ Standarization is especially important for machine learning models that rely on distance metrics (eg, K-NN, SVM) or gradient based optimizations (eg logical regression, neural networks)

Z-Score Standarization
→ Z score standarization is a method to standarize data using the following formula

$$Z = \frac{x - \mu}{\sigma}$$

Where
x: the data point to be standarized

→ μ : mean of the dataset
→ σ : the SD of the dataset
→ Z : the standarized value
Thus transformation results in a dataset with
- Mean (μ) = 0
- Standard Deviation (σ) = 1

Example
Suppose we have a dataset of student's test scores
[70, 80, 90, 85, 75]

1) Calculate the Mean (μ)

$$\mu = \frac{70+80+90+85+75}{5} = 80$$

2) Calculate the SD (σ)

$$\sigma = \sqrt{\frac{\Sigma(x_i-\mu)^2}{N}} = \sqrt{\frac{(70-80)^2+(80-80)^2+(90-80)^2+(85-80)^2+(75-80)^2}{5}}$$

$$= \sqrt{50} \cong 7.07$$

3) Standardize each value
For $x = 70$

$$Z = \frac{70-80}{7.07} \approx -1.41$$

For $x = 80$

$$Z = \frac{80-80}{7.07} \approx 0$$

For $x = 85$, $x = 75$, $x = 90$
$Z_{85} = 0.71$, $Z_{75} = -0.71$, $Z_{90} \cong 1.41$

Z-Score Standardized Dataset
[-1.41, 0, 1.41, 0.71, -0.71]

Q-13 What is Chi-square test? Why it is necessary in data analysis?

Ans The Chi-square test is a statistical method used to determine whether there is a significant association between categorical variables
→ The test evaluates whether the differences between observed and expected data are due to chance or a relationship between the variables
The Chi-square test is necessary in data analysis for several reasons
1) Identifying Relationships Between Variables
→ It helps determine if there is a dependency between variables (eg education level and income group)
2) Validating Hypotheses
→ Provides statistical evidence to accept or reject a null hypotheses
3) Detecting Distribution Patterns
→ Verifies whether an observed distribution matches an expected one (eg survey responses vs expected tools)
4) Making Data-Driven Decisions
→ Helps analysts and bussiness understand relationships, guiding decisions in marketing, operations or research

Q-14 Describe Sampling along with its types in detail with suitable example

Ans Sampling is the process of selecting a subset (sample) from a larger population to draw conclusions about

the entire population. Sampling is essential when analyzing the entire population is impractical, time-consuming and costly

0 Types of Sampling
1) Probability Sampling
2) Non-Probability Sampling

1) Probability Sampling
→ In probability sampling, every member of the population has a known, non-zero chance of being selected
→ This method ensures the sample is representative of the population, reducing bias.

2) Non-Probability Sampling
→ In non-probability sampling, not all individuals have a chance of being included. This approach is easier and quicker but may lead to bias

Example: Sampling Types in Practice
Scenario: A company wants feedback from its 10,000 employees
1) Simple Random Sampling: Randomly select 500 employees using software
2) Stratified Sampling: Divide employees into departments (eg HR, IT, Sales) and sample proportionally
3) Cluster Sampling: Randomly pick 10 office locations and survey all employees there
4) Convenience Sampling: Survey employees available at the cafeteria during lunch
5) Snowball Sampling: Ask surveyed employees to recommend colleagues to participate

Q.5 Write a Python program to demonstrate the concepts of Skewness and Kurtosis

Ans
```
import numpy as np
From scipy.stats import skew, kurtosis

# Sample data : Normal and Skewed kurtosis
data_normal = [10, 12, 15, 18, 20, 25, 30]  # Symmetric - data
data_skewed = [10, 12, 15, 18, 20, 50, 100]  # Right skewed data

# Calculate Skewness and Kurtosis
normal_skewness = skew(data_normal)
normal_kurtosis = kurtosis(data_skewed)

# Print Results
print("Normal data:")
print(f"Skewness : {normal_skewness :.4f}")
print(f"Kurtosis : {normal_kurtosis :.4f}\n")

print("Skewed data:")
print(f"Skewness : {skewed_skewness :.4f}")
print(f"Kurtosis : {skewed_kurtosis :.4f}")
```

Output
```
Normal Data:
Skewness : 0.0000
Kurtosis : -1.2615

Skewed data
Skewness : 1.5432
Kurtosis : 1.4312
```

Q-16 Explain any three functions from Sakit Learn

Ans 1) train_test_split
Purpose: Splits a dataset into training and testing subsets
Usage: This function is essential for evaluating the performance of a machine learning model. By dividing the dataset, you can train the model on one portion and test it on unseen data to avoid overfitting

2) Fit
Purpose: Fits a machine learning model to the training data
Usage: The function trains the model by learning from the patterns in the training data

3) Predict
Purpose: Makes prediction based on the fitted model
Usage: After training a model using fit, you can use predict to estimate outcomes for new or unseen data

Q-17 Explain the use of skew() and kurtosis() function

Ans 1) skew() function
The skew() function measures the asymmetry of the probability distribution of a dataset
Key Points:
→ Symmetric Distribution: Skewness = 0 (eg normal distribution)
→ Positive Skewness: Long tail on the right side (values are concentrated on the left)
→ Negative Skewness: Long tail on the left side (values are concentrated on the right)
Syntax:
scipy.stats.skew(a, axis=0, bias=True)

2 Kurtosis() Function
The Kurtosis() Function measures the "tailedness" or ... in the data distribution
Key Points
→ Normal Distribution : Kurtosis = 0
→ Positive Kurtosis (LeptoKurtic) : Heavy-tailed distribution with more extreme outliers
→ Negative Kurtosis (PlatyKurtic) : Light-tailed distribution with fewer extreme outliers

Q-18 Explain Classification and clustering class of Scikit-learn

Ans 1) Classification (Supervised Learning)
→ Classification is a supervised learning task where the goal is to predict a discrete label (category) for a given input based on labeled training data. The model learns from the input-output pairs to classify new, unseen data.
o Key Points
1) Input: Labeled dataset (Features and Corresponding target labels)
2) Output: A category or class label for new data points
Examples
o Spam mail detection
o Image recognition
o Disease recognition

(2) Clustering (Unsupervised Learning)
→ Clustering is an unsupervised learning task where the goal is to group similar data points together into clusters without using labeled data. The algorithm identifies patterns or structures in the data
o Key Points
1) Input: Unlabeled dataset (only features)

2) Output: Cluster labels indicating group membership

Example

1) Customer Segmentation
2) Document Segmentation
3) Social network analysis