Note: In Web Technology, we did not complete slips 24-30 because they pertain to files and databases. Therefore, there is no need to cover the same topics in Data Science.

Slip 1

A) Write a Python program to create a Pie plot to get the frequency of the three species of the Iris data (Use iris.csv)

→

```
import pandas as pd
import matplotlib.pyplot as plt
iris=pd.read_csv("iris.csv")
s_c=iris['species'].value_counts()
plt.figure()
plt.pie(s_c,labels=s_c.index)
plt.title("freq")
plt.show()
```

B) Write a Python program to view basic statistical details of the data.(Use wineequality-red.csv)

—>

```
import pandas as pd
w_d=pd.read_csv("winequality-red.csv")
stat=w_d.describe()
print(stat)
```

Slip 2

A) Write a Python program for Handling Missing Value. Replace missing value of salary, age column with mean of that column.(Use Data.csv file).]

→

```
import pandas as pd

# Create DataFrame and fill missing values
data = pd.DataFrame({'salary': [50000, 60000, None, 45000, None], 'age': [25, None, 30, None, 40]})
print("Missing before:", data.isnull().sum())
data.fillna(data.mean(), inplace=True)
print("Missing after:", data.isnull().sum())

# Display the DataFrame
print(data)
```

B) Write a Python program to generate a line plot of name Vs salary

→

```
import matplotlib.pyplot as plt
data = {
'name': ['John', 'Alice', 'Bob', 'Eve', 'Charlie'],
'salary': [50000, 60000, 75000, 55000,
80000]
}
df = pd.DataFrame(data)

plt.plot(df['name'], df['salary'])
plt.title('Name vs Salary')
plt.xlabel('Name')
plt.ylabel('Salary')
plt.show()
```

C) Download the heights and weights dataset and load the dataset from a given csv file into a dataframe. Print the first, last 10 rows and random 20 rows also display the shape of the dataset.

→

```
import pandas as pd

df = pd.read_csv('SOCR-HeightWeight.csv')

print("First 10 rows:\n", df.head(10))
print("\nLast 10 rows:\n", df.tail(10))
print("\nRandom 20 rows:\n", df.sample(20))
print("\nShape:\n", df.shape)
```

Slip 3

A.) Write a Python program to create box plots to see how each feature i.e. Sepal Length, Sepal Width, Petal Length, Petal Width are distributed across the three species. (Use iris.csv dataset)

→

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

# Load the Iris dataset
df = pd.read_csv('iris.csv')

# Create box plots for each feature
df.boxplot(column=['sepal_length', 'sepal_width', 'petal_length', 'petal_width'], by='species')
plt.title('Box Plots of Iris Features by Species')
plt.suptitle('')
plt.show()
```

B) Write a Python program to view basic statistical details of the data (Use Heights and Weights Dataset)

→

```
import pandas as pd
w_d=pd.read_csv("SOCR-HeightWeight.csv")
stat=w_d.describe()
print(stat)
```

Slip 4

A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate colour, labels and styling options.

→

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

# Generate random data
data = np.random.randint(1, 101, size=50)

# Plot charts
plt.figure();
plt.plot(data);
plt.title("Line Chart");
plt.show()
plt.figure();
plt.scatter(range(len(data)), data);
plt.title("Scatter Chart");
```

```
plt.show()
plt.figure();
plt.hist(data, bins=10);
plt.title("Histogram Chart");
plt.show()
plt.figure();
plt.boxplot(data);
plt.title("Box Plot");
plt.show()
```

B) Write a Python program to print the shape, number of rows-columns, data types, feature names and the description of the data(Use User_Data.csv)

→

```
import pandas as pd

# Load the dataset
df = pd.read_csv('User_Data.csv')

# Print data information
print("Shape:", df.shape)
print("Number of Rows:", df.shape[0])
print("Number of Columns:", df.shape[1])
print("Data Types:\n", df.dtypes)
print("Feature Names:", df.columns.tolist())
print("Description:\n", df.describe())
```

Slip 5

A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate colour, labels and styling options

→

Follow slip 4. A

B) Write a Python program to print the shape, number of rows-columns, data types, feature names and the description of the data(Use User_Data.csv)

→

Follow slip 4.B

Slip 6

A) Write a Python program for Handling Missing Value. Replace missing value of salary, age column with mean of that column.(Use Data.csv file).
—>
Follow slip 2.A

B) Write a Python program to generate a line plot of name Vs salary
→
Follow slip 2.B

C) Download the heights and weights dataset and load the dataset from a given csv file into a dataframe. Print the first, last 10 rows and random 20 rows also display the shape of the dataset.
→ **follow slip 2.C**

Slip 7

Q.2) Write a Python program to perform the following tasks :
a. Apply OneHot coding on the Country column.
b. Apply Label encoding on purchased column
(Data.csv has two categorical columns: the country column, and the purchased column).
→

```
import pandas as pd

# Load the dataset
df = pd.read_csv('Data.csv')

# Apply One-Hot Encoding on Country column
df = pd.get_dummies(df, columns=['Country'], drop_first=True)

# Apply Label Encoding on Purchased column using simple function
df['Purchased'] = df['Purchased'].map({'No': 0, 'Yes': 1})

# Display the transformed DataFrame
print(df.head())
```

Slip 8

A.)Write a program in python to perform following task :
Standardising Data (transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1) (Use winequality-red.csv)

→

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
# Step 1: Load the dataset from a CSV file
df = pd.read_csv('winequality-red.csv')

# Step 2: Initialise the StandardScaler to standardise the data

scaler = StandardScaler()

# Step 3: Standardise the dataset
# The fit_transform method scales the data to have a mean of 0 and standard deviation of 1

df_scaled = pd.DataFrame(scaler.fit_transform(df), columns=df.columns)
# Step 4: Display the first few rows of the standardised DataFrame
print("Standardised Data:\n", df_scaled.head())
```

Slip 9

A) Generate a random array of 50 integers and display them using a line chart, scatter plot. Apply appropriate colour, labels and styling options.

→

Follow slip 4.A

B) Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart.

→

```
import matplotlib.pyplot as plt

# Step 1: Create lists for subjects and marks
subjects = ['Maths', 'Science', 'English', 'History']
marks = [85, 90, 75, 80]

# Step 2: Create a pie chart
plt.pie(marks, labels=subjects, autopct='%1.1f%%')

# Step 3: Add a title
plt.title('Marks Distribution by Subject')

# Step 4: Show the pie chart
plt.show()
```

C) Write a program in python to perform following task (Use winequality-red.csv) Import Dataset and do the followings:

a) Describing the dataset

b) Shape of the dataset

c) Display first 3 rows from dataset

→

```
import pandas as pd
# Step 1: Load the dataset
df = pd.read_csv('winequality-red.csv')
# Step 2: Describe the dataset
print("Dataset Description:\n", df.describe())
# Step 3: Shape of the dataset
print("\nShape of the dataset:", df.shape)
# Step 4: Display the first 3 rows
print("\nFirst 3 rows of the dataset:\n", df.head(3))
```

Slip 10

A) Write a python program to Display column-wise mean, and median for SOCRHeightWeight dataset.

→

```
import pandas as pd
# Load the dataset
df = pd.read_csv('SOCR-HeightWeight.csv')
# Calculate mean and median
mean_values = df.mean()
median_values = df.median()
# Display results
print("Column-wise Mean:\n", mean_values)
print("\nColumn-wise Median:\n", median_values)
```

B) Write a python program to compute sum of Manhattan distance between all pairs of point

→

```
# Sample list of points (x, y)
points = [(1, 2), (3, 5), (6, 1), (2, 4)]

# Calculate the sum of Manhattan distances
total_distance = sum(
    abs(x1 - x2) + abs(y1 - y2)
    for i, (x1, y1) in enumerate(points)
    for j, (x2, y2) in enumerate(points)
    if i < j
)
# Display the total Manhattan distance
print("Sum of Manhattan distances:", total_distance)
```

Slip 11

A) Write a Python program to create a Pie plot to get the frequency of the three species of the Iris data (Use iris.csv)

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

# Load the dataset
df = pd.read_csv('iris.csv')

# Count species frequencies
species_counts = df['species'].value_counts()

# Create a pie plot with clear variables
plt.pie(species_counts.values, labels=species_counts.index, autopct='%1.1f%%')
plt.title('Frequency of Iris Species')
plt.show()
```

B) Write a Python program to view basic statistical details of the data.(Use wineequality-red.csv)

→

Follow slip 1

Slip 12

B) Write a Python program to create a data frame containing column name, salary, department add 10 rows with some missing and duplicate values to the data frame. Also drop all null and empty values. Print the modified data frame.

→

```
import pandas as pd

# Create a DataFrame with some missing and duplicate values
data = {
    'Name': ['Alice', 'Bob', 'Alice', None],
    'Salary': [70000, None, 70000, 60000],
    'Department': ['HR', 'Finance', 'HR', None]
}

df = pd.DataFrame(data)

# Drop all null values
df.dropna(inplace=True)

# Print the modified DataFrame
print(df)
```

A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate colour, labels and styling options.

→

Follow slip 4.A

Slip 13

A) Write a Python program to create a graph to find the relationship between the petal length and petal width.(Use iris.csv dataset)

→

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

# Load the dataset
df = pd.read_csv('iris.csv')

# Create a scatter plot
plt.scatter(df['petal_length'], df['petal_width'])

# Add labels and title
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.title('Relationship between Petal Length and Petal Width')

# Show the plot
plt.show()
```

B) Write a Python program to find the maximum and minimum value of a given flattened array

→

```
import numpy as np

# Create a flattened array
array = np.array([3, 1, 7, 4, 9, 2, 5])

# Find maximum and minimum values
max_value = np.max(array)
min_value = np.min(array)

# Display the results
print("Maximum value:", max_value)
print("Minimum value:", min_value)
```

Slip 14

A) Write a Python NumPy program to compute the weighted average along the specified axis of a given flattened array.

→

```
import numpy as np

# Create a flattened array and weights
array = np.array([3, 1, 7, 4, 9])
weights = np.array([0.1, 0.2, 0.3, 0.2, 0.2])

# Compute the weighted average
weighted_average = np.average(array, weights=weights)

# Display the result
print("Weighted Average:", weighted_average)
```

B) Write a Python program to view basic statistical details of the data (Use advertising.csv)

→

Follow slip 3.B

Slip 15

A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate colour, labels and styling options.

→

Follow slip 4

B) Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart.

→

Follow Slip 9

Slip 16

A) Write a python program to create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart and bar chart.
→Same as a slip 9… only bar chart is added

```
import matplotlib.pyplot as plt

# Step 1: Create lists for subjects and marks
subjects = ['Maths', 'Science', 'English', 'History']
marks = [85, 90, 75, 80]

# Step 2: Create a pie chart
plt.pie(marks, labels=subjects, autopct='%1.1f%%')
plt.title('Marks Distribution by Subject')
plt.show()  # Show the pie chart

# Step 3: Create a bar chart
plt.bar(subjects, marks, colour='skyblue')
plt.xlabel('Subjects')
plt.ylabel('Marks')
plt.title('Marks Distribution by Subject (Bar Chart)')
plt.show()  # Show the bar chart
```

B) Write a python program to create a data frame for students' information such as name, graduation percentage and age. Display average age of students, average of graduation percentage.
→

```
import pandas as pd

# Create a DataFrame for students' information
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],
    'Graduation Percentage': [85.5, 90.0, 78.0, 92.5, 88.0],
    'Age': [21, 22, 23, 21, 22]
}

df = pd.DataFrame(data)

# Calculate average age and graduation percentage
average_age = df['Age'].mean()
average_percentage = df['Graduation Percentage'].mean()
# Display the results
print("Average Age of Students:", average_age)
print("Average Graduation Percentage:", average_percentage)
```

Slip 17

A) Write a Python program to draw scatter plots to compare two features of the iris dataset

→

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

# Load the iris dataset
df = pd.read_csv('iris.csv')

# Scatter plot for petal length vs petal width
plt.scatter(df['petal_length'], df['petal_width'], c='blue', alpha=0.5)

# Add labels and title
plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.title('Petal Length vs Petal Width')

# Display the plot
plt.show()
```

B) Write a Python program to create a data frame containing columns name, age , salary, department . Add 10 rows to the data frame. View the data frame

→

```
import pandas as pd

# Create a DataFrame with columns
data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, 30, 22],
    'Salary': [50000, 60000, 45000],
    'Department': ['HR', 'Finance', 'IT']
}

df = pd.DataFrame(data)

# Display the DataFrame
print(df)
```

Slip 18

A) Write a Python program to create box plots to see how each feature i.e. Sepal Length, Sepal Width, Petal Length, Petal Width are distributed across the three species. (Use iris.csv dataset)

→

Follow slip 3.A

B) Use the heights and weights dataset and load the dataset from a given csv file into a dataframe. Print the first, last 5 rows and random 10 row

→

```
import pandas as pd

# Load the heights and weights dataset from a CSV file
df = pd.read_csv('SOCR-HeightWeight.csv')  # Replace with your actual CSV file path

# Print the first 5 rows
print("First 5 rows:")
print(df.head())

# Print the last 5 rows
print("\nLast 5 rows:")
print(df.tail())

# Print 10 random rows
print("\nRandom 10 rows:")
print(df.sample(10))
```

Slip 19

A. To create a dataframe containing columns name, age and percentage. Add 10 rows to the dataframe. View the data frame.

→ **Follow Slip 17.B**

B. To print the shape, number of rows-columns, data types, feature names and the description of the data

→

```
import pandas as pd

# Create a sample DataFrame
data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [20, 22, 19],
    'Percentage': [85.5, 90.0, 78.0]
```

```
}

df = pd.DataFrame(data)

# Print the shape (number of rows and columns)
print("Shape of the DataFrame:", df.shape)

# Print the number of rows and columns
print("Number of rows:", df.shape[0])
print("Number of columns:", df.shape[1])

# Print data types of each column
print("\nData types:")
print(df.dtypes)

# Print feature names (column names)
print("\nFeature names:")
print(df.columns.tolist())

# Print description of the data
print("\nDescription of the data:")
print(df.describe(include='all'))
```

C. To Add 5 rows with duplicate values and missing values. Add a column 'remarks' with empty values. Display the data.

→

```
import pandas as pd
import numpy as np

# Create a sample DataFrame
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],
    'Age': [20, 22, 19, 22, np.nan],  # Adding a missing value
    'Percentage': [85.5, 90.0, 78.0, 90.0, 85.5]  # Duplicate value
}

df = pd.DataFrame(data)

# Adding 5 rows with duplicate values and a missing value
duplicate_data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'Alice', 'Eve'],  # Duplicates
    'Age': [20, 22, 19, np.nan, 22],  # Adding a missing value
    'Percentage': [85.5, 90.0, 78.0, 85.5, 90.0]  # Duplicates
}
```

```
# Create a DataFrame for the duplicate data
duplicate_df = pd.DataFrame(duplicate_data)

# Concatenate the original DataFrame with the duplicate DataFrame
df = pd.concat([df, duplicate_df], ignore_index=True)

# Add a 'remarks' column with empty values
df['Remarks'] = ''

# Display the DataFrame
print(df)
```

Slip 20

A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate colour, labels and styling options.

→

Slip 3

Q.2 B) Add two outliers to the above data and display the box plot

→

Not understand

Slip 21

A) Import dataset "iris.csv". Write a Python program to create a Bar plot to get the frequency of the three species of the Iris data.

→

Follow slip 11.A

Q.2 B)Write a Python program to create a histogram of the three species of the Iris data.

→

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

# Load the Iris dataset
df = pd.read_csv('iris.csv')  # Make sure to use the correct path to your CSV file

# Create a histogram for each species
```

```
plt.figure(figsize=(10, 6))

# Plot histogram for each species
for species in df['species'].unique():
    subset = df[df['species'] == species]
    plt.hist(subset['sepal_length'], bins=10, alpha=0.5, label=species)

# Add titles and labels
plt.title('Histogram of Sepal Length for Different Iris Species')
plt.xlabel('Sepal Length')
plt.ylabel('Frequency')
plt.legend(title='Species')
plt.grid(True)

# Show the histogram
plt.show()
```

Slip 22

A.) Dataset Name: winequality-red.csv
Write a program in python to perform following tasks
a. Rescaling: Normalised the dataset using MinMaxScaler class
b. Standardising Data (transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1)
c. Normalising Data (rescale each observation to a length of 1 (a unit norm). For this, use the Normalizer class.)
→
One in all

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer

# Load the dataset
df = pd.read_csv('winequality-red.csv')

# a. Rescaling using MinMaxScaler
normalized_df = pd.DataFrame(normalized_data)

# b. Standardising the data
standardized_df = pd.DataFrame(standardized_data)

# c. Normalising the data
normalized_length_df = pd.DataFrame(normalized_length_data)

# Display the results
```

```
print("Normalised Data (MinMaxScaler):\n", normalized_df.head())
print("\nStandardized Data:\n", standardized_df.head())
print("\nNormalized Length Data:\n", normalized_length_df.head())
```

Slip 23

A.)Dataset Name: winequality-red.csv
Write a program in python to perform the following task a. Rescaling: Normalised the dataset using MinMaxScaler class
b. Standardising Data (transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1)
c. Binarizing Data using we use the Binarizer class (Using a binary threshold, it is possible to transform our data by marking the values above it 1 and those equal to or below it, 0)
→
Same as slip 22 only c point change

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer,Binarizer

# Load the dataset
df = pd.read_csv('winequality-red.csv')

# a. Rescaling using MinMaxScaler
normalized_df = pd.DataFrame(normalized_data)

# b. Standardising the data
standardized_df = pd.DataFrame(standardized_data)

# c. Normalising the data
normalized_length_df = pd.DataFrame(normalized_length_data)

# d. Binarizing the data
binarized_df = pd.DataFrame(binarized_data, columns=df.columns)

# Display the results
print("Normalised Data (MinMaxScaler):\n", normalized_df.head())
print("\nStandardized Data:\n", standardized_df.head())
print("\nNormalized Length Data:\n", normalized_length_df.head())
print("\nBinarized Data:\n", binarized_df.head())
```

Slip 24
Q.2 A) Import dataset “iris.csv”. Write a Python program to create a Bar plot to get the frequency of the three species of the Iris data. [10] Q.2 B) Write a Python program to create a histogram of the three species of the Iris data

Slip 25
Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate colour, labels and styling options. [10] Q.2 B) Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart.

Slip 26
Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate colour, labels and styling options. [10] 2.Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in the bar chart.

Slip 27
Q.2) Create a dataset data.csv having two categorical columns (the country column, and the purchased column). [15] a. Apply OneHot coding on the Country column. b. Apply Label encoding on purchased column

Slip 28
Q.2) Write a Python program [15] 1. To create a dataframe containing columns name, age and percentage. Add 10 rows to the dataframe. View the data frame. 2. To print the shape, number of rows-columns, data types, feature names and the description of the data. 3. To

view basic statistical details of the data. 4. To Add 5 rows with duplicate values and missing values. Add a column ‘remarks’ with empty values. Display the data.

Slip 29

Q.2) Create a dataset data.csv having two categorical columns (the country column, and the purchased column). [15] 1. Apply OneHot coding on the Country column. 2. Apply Label encoding on purchased column

Slip 30

Q.2) Write a python program to [15] a. Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate colour, labels and styling options. b. Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in the bar chart.

Slip 23