Slip1:

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 Iris dataset from the CSV file

data = pd.read_csv(r'D:\DS_Final\exam\TYDATA\Iris.csv')

# Calculate the frequency of each species

sp_cnt= data['Species'].value_counts()

# Create a pie chart

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

plt.pie(sp_cnt, labels=sp_cnt.index, autopct='%1.1f%%', startangle=140)

plt.title('Iris Species Frequency')

# Display the pie chart

plt.show()
```

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

```
import pandas as pd

# Load the winequality-red dataset from the CSV file

data = pd.read_csv('D:\DS_Final\exam\TYDATA\winequality-red.csv')

# Display basic statistical details

statistical_details = data.describe()

# Print the statistical details to the console

print(statistical_details)
```

Slip2:

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
# Load the dataset from the CSV file
data = pd.read_csv(r'D:\DS_Final\exam\TYDATA\data.csv')
# Calculate the mean of the "salary" and "age" columns
mean_salary = data['Salary'].mean()
mean_age = data['Age'].mean()
# Replace missing values in the "salary" and "age" columns with their respective means
data['Salary'].fillna(mean_salary, inplace=True)
data['Age'].fillna(mean_age, inplace=True)
# Display the first few rows of the modified dataset
print(data)
```

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

```
import pandas as pd
import matplotlib.pyplot as plt
# Load the dataset from the CSV file
data = pd.read_csv(r'D:\DS_Final\exam\TYDATA\data.csv')
# Sort the data by "name" (optional, if not already sorted)
data = data.sort_values(by='Name')

# Create a line plot of "name" versus "salary"
plt.figure(figsize=(12, 6))
plt.plot(data['Name'], data['Salary'], marker='o', linestyle='-', color='b')
plt.xlabel('Name')
plt.ylabel('Salary')
plt.title('Line Plot of Name vs Salary')
plt.xticks(rotation=90)  # Rotate the x-axis labels for better readability
plt.grid(True)
# Show the plot
plt.show()
```

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 shape of the dataset.

```
import pandas as pd
import numpy as np
# Load the dataset from the CSV file
data = pd.read_csv(r'D:\DS_Final\exam\TYDATA\weight-height.csv')
# Print the first 10 rows
print("First 10 rows:")
print(data.head(10))
# Print the last 10 rows
print("\nLast 10 rows:")
print(data.tail(10))
print("\nRandom 20 rows:")
random_20_rows = data.sample(20)
print(random_20_rows)
```

```
# Display the shape of the dataset

print("\nShape of the dataset:", data.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 from the CSV file
data = pd.read_csv(r'D:\DS_Final\exam\TYDATA\Iris.csv')

# Create a box plot for each feature across the three species
plt.figure(figsize=(12, 8))

# Sepal Length
plt.subplot(2, 2, 1)
data.boxplot(column='SepalLength', by='Species')
plt.title('Sepal Length by Species')
# Sepal Width
plt.subplot(2, 2, 2)
data.boxplot(column='SepalWidth', by='Species')
plt.title('Sepal Width by Species')
# Petal Length
plt.subplot(2, 2, 3)
data.boxplot(column='PetalLength', by='Species')
plt.title('Petal Length by Species')

# Petal Width
plt.subplot(2, 2, 4)
data.boxplot(column='PetalWidth', by='Species')
plt.title('Petal Width by Species')
# Show the box plots
plt.show()
```

Q.2 B) Write a Python program to view basic statistical details of the data (Use Heights and Weights Dataset)

```
import pandas as pd

data = pd.read_csv('D:\DS_Final\exam\TYDATA\weight-height.csv')

print("First few rows of the dataset:")

print(data.head())

print("\nBasic Statistical Details:")

print(data.describe())

print("\nAdditional Information:")

print(data.info())
```

Slip 4:

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 color, labels and styling options. [10]

```
import numpy as np
import matplotlib.pyplot as plt
data = np.random.randint(0, 100, 50)
# Line chart
plt.subplot(2,2,1)
plt.plot(data, marker='o')
plt.title('Line Chart')

# Scatter plot
plt.subplot(2,2,2)
plt.scatter(range(50), data, marker='x')
plt.title('Scatter Plot')

# Histogram
plt.subplot(2,2,3)
plt.hist(data, bins=10)
plt.title('Histogram')

# Box plot
plt.subplot(2,2,4)
plt.boxplot(data, vert=True)
plt.title('Box Plot')

# Display the plots
plt.tight_layout()
plt.show()
```

Q.2 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

data = pd.read_csv(r'D:\DS_Final\exam\TYDATA\User_Data.csv')

# Print the shape (number of rows and columns)

print("Shape of the dataset:", data.shape)

# Print the number of rows and columns

num_rows, num_cols = data.shape

print(f"Number of rows: {num_rows}")

print(f"Number of columns: {num_cols}")


# Print data types and feature names

print("\nData types and feature names:")
print(data.dtypes)

print("\nFeature names:")

print(data.columns)


# Print a description of the data

print("\nDescription of the data:")

print(data.describe())
```

Slip 5:

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 color, labels and styling options. [10]

Please refer from above program …Already Solved question

Q.2 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

data = pd.read_csv(r'D:\DS_Final\exam\TYDATA\User_Data.csv')

# Print the shape (number of rows and columns)

print("Shape of the dataset:", data.shape)

# Print the number of rows and columns

num_rows, num_cols = data.shape

print(f"Number of rows: {num_rows}")

print(f"Number of columns: {num_cols}")

# Print data types and feature names

print("\nData types and feature names:")

print(data.dtypes)

print("\nFeature names:")

print(data.columns)

# Print a description of the data

print("\nDescription of the data:")

print(data.describe())
```

Slip 6:

Q.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). [5]

```
import pandas as pd
data = pd.read_csv(r'D:\DS_Final\exam\TYDATA\data.csv')
```

```
# Calculate the mean of the "salary" and "age" columns
mean_salary = data['Salary'].mean()
mean_age = data['Age'].mean()

# Replace missing values in the "salary" and "age" columns with their respective means
data['Salary'].fillna(mean_salary, inplace=True)
data['Age'].fillna(mean_age, inplace=True)

# Display the first few rows of the modified dataset
print(data)
```

Q.2 B) Write a Python program to generate a line plot of name Vs salary [5]

```
import pandas as pd

import matplotlib.pyplot as plt

# Load the dataset from the CSV file

data = pd.read_csv(r'D:\DS_Final\exam\TYDATA\data.csv')

# Sort the data by "name" (optional, if not already sorted)

data = data.sort_values(by='Name')

# Create a line plot of "name" versus "salary"

plt.figure(figsize=(12, 6))

plt.plot(data['Name'], data['Salary'], marker='o', linestyle='-', color='b')

plt.xlabel('Name')

plt.ylabel('Salary')

plt.title('Line Plot of Name vs Salary')

plt.xticks(rotation=90)  # Rotate the x-axis labels for better readability

plt.grid(True)

plt.show()
```

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

```
data = pd.read_csv(r'D:\DS_Final\exam\TYDATA\weight-height.csv')

print("First 10 rows:")

print("\nLast 10 rows:")

print(data.tail(10))

print("\nRandom 20 rows:")

print(data.sample(20))

print("\nShape of the dataset:", data.shape)
```

Slip 7:

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

```
import pandas as pd
# Load the dataset from the CSV file (replace 'Data.csv' with the actual file path or URL)
data = pd.read_csv(r'D:\DS_Final\exam\TYDATA\data.csv')
# Apply One-Hot encoding to the 'Country' column
data = pd.get_dummies(data, columns=['Country'], prefix='Country')
# Apply Label encoding to the 'Purchased' column
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
data['Purchased'] = label_encoder.fit_transform(data['Purchased'])
# Print the updated DataFrame
print(data)
```

Slip no8:

Q.2) Write a program in python to perform following task : [15]
Standardizing 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
data = pd.read_csv(r'D:\DS_Final\exam\TYDATA\winequality-red.csv')
# Extract the features (exclude the target variable if present)
feature = data.drop(columns=['quality'])
# Standardize the data
scaler = StandardScaler()
std_data = scaler.fit_transform(feature)
# Create a DataFrame from the standardized data
std_df = pd.DataFrame(std_data, columns=feature.columns)
# Display the first few rows of the standardized data and its summary statistics
print("First few rows of standardized data:")
print(std_df.head())


print("\nSummary statistics of standardized data:")
print(std_df.describe())
```

Slip no 9:

Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot. Apply appropriate color, labels and styling options. [5]

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

data = np.random.randint(0, 100, 50)

# Line chart
plt.subplot(2,2,1)
plt.plot(data, marker='o')
plt.title('Line Chart')

# Scatter plot
plt.subplot(2,2,2)
plt.scatter(range(50), data, marker='x')
plt.title('Scatter Plot')

# Histogram
plt.subplot(2,2,3)
plt.hist(data, bins=10)
plt.title('Histogram')

# Box plot
plt.subplot(2,2,4)
plt.boxplot(data, vert=True)
plt.title('Box Plot')

# Display the plots
plt.tight_layout()
plt.show()
```

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. [5]

```
import matplotlib.pyplot as plt

# Sample subject names and marks obtained
subjects = ["Math", "Science", "History", "English", "Art"]
marks = [90, 80, 75, 88, 95]

# Create a pie chart
plt.figure(figsize=(6, 6))
plt.pie(marks, labels=subjects, autopct='%1.1f%%', startangle=140)

# Add a title
plt.title("Subject-wise Marks")

# Display the pie chart
plt.show()
```

Q.2 C) Write a program in python to perform following task (Use winequality-red.csv) [5]
Import Dataset and do the followings:
a) Describing the dataset
b) Shape of the dataset
c) Display first 3 rows from dataset

Please refer from above program …Already Solved question

Slip no 10:

Q.2 A) Write a python program to Display column-wise mean, and median for SOCR-HeightWeight dataset. [10]
Please refer from above program …Already Solved question

Q.2 B) Write a python program to compute sum of Manhattan distance between all pairs of points.

```
        def manhattan_distance(point1, point2):

        return abs(point1[0] - point2[0]) + abs(point1[1] - point2[1])


        def sum_manhattan_distances(points):

         total_distance = 0

         n = len(points)

           for i in range(n):

                for j in range(i + 1, n):

                        total_distance += manhattan_distance(points[i], points[j])

                 return total_distance

points = [(1, 2), (4, 6), (3, 8), (7, 1)]

total_distance = sum_manhattan_distances(points)

print("Sum of Manhattan distances between all pairs of points:", total_distance)
```

Slip no 11:

Q.2 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) [10]

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

data = pd.read_csv(r'D:\DS_Final\exam\TYDATA\Iris.csv')

# Count the frequency of each species
species_counts = data['Species'].value_counts()

# Create a pie plot
plt.figure(figsize=(6, 6))
plt.pie(species_counts, labels=species_counts.index, autopct='%1.1f%%', startangle=140)

# Add a title
```

```
plt.title("Frequency of Iris Species")

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

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

```
import pandas as pd
data = pd.read_csv('D:\DS_Final\exam\TYDATA\weight-height.csv')
print("First few rows of the dataset:")
print(data.head())
print("\nBasic Statistical Details:")
print(data.describe())
print("\nAdditional Information:")
print(data.info())
```

Slip no 12:

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 color, labels and styling options. [10]

Please refer from above program …Already Solved question

Q.2 B) Write a Python program to create 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
import numpy as np
# Create an initial DataFrame
data = {
    "name": ["Alice", "Bob", "Charlie", "David", "Eve"],
    "salary": [50000, 60000, 70000, 55000, 62000],
    "department": ["HR", "Finance", "Engineering", "HR", "Engineering"]
}
df = pd.DataFrame(data)


# Adding 10 rows with missing and duplicate values
new_data = {
    "name": ["Alice", "Eve", "Frank", "Bob", "Grace", "Alice", "Bob", "Hannah", "David", "Eve"],
    "salary": [50000, np.nan, 75000, 60000, np.nan, 50000, 60000, np.nan, 55000, 62000],
    "department": ["HR", "Engineering", "Marketing", "Finance", "", "HR", "Finance", "", "HR", "Engineering"]

}


df = df.append(pd.DataFrame(new_data), ignore_index=True)


# Drop rows with null and empty values

df = df.dropna(subset=["name", "salary", "department"]).dropna(how="all")


# Print the modified DataFrame

print(df)
```

Slip no 13:

Q.2 A) Write a Python program to create a graph to find relationship between the petal length and petal width.(Use iris.csv dataset) [10]

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

# Load the Iris dataset
iris_data = pd.read_csv(r'D:\DS_Final\exam\TYDATA\Iris.csv')

# Extract petal length and petal width columns
petal_length = iris_data['PetalLength']
petal_width = iris_data['PetalWidth']

# Create a scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(petal_length, petal_width, c='b', marker='o', label='Petal Length vs. Petal Width')

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

# Show the plot
plt.legend()
plt.grid(True)
plt.show()
```

Q.2 B) Write a Python program to find the maximum and minimum value of a given flattened array.

```
import numpy as np

# Create a sample flattened array (replace this with your own array)

flattened_array = np.array([5, 12, 8, 15, 3, 10, 25, 1, 7])

# Find the maximum and minimum values
```

```
max_value = np.max(flattened_array)

min_value = np.min(flattened_array)

# Print the results

print(f"Maximum value: {max_value}")

print(f"Minimum value: {min_value}")
```

Slip no 14:

Q. 2 A) Write a Python NumPy program to compute the weighted average along the specified axis of a given flattened array. [10]

```
import numpy as np
flattened_array = np.array([5, 12, 8, 15, 3, 10, 25, 1, 7])
weights = np.array([0.1, 0.2, 0.15, 0.1, 0.05, 0.2, 0.1, 0.05, 0.05])
weighted_average = np.average(flattened_array, weights=weights)
print(f"Weighted Average: {weighted_average}")
```

Q. 2 B) Write a Python program to view basic statistical details of the data (Use advertising.csv)

Please refer from above program …Already Solved question

Slip no 15:

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 color, labels and styling options. [10]

Please refer from above program …Already Solved question

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.

```
import matplotlib.pyplot as plt

subject_names = ["Math", "Science", "History", "English", "Art"]

marks_obtained = [90, 85, 78, 92, 70]

plt.figure(figsize=(8, 8))

plt.pie(marks_obtained, labels=subject_names, autopct='%1.1f%%', startangle=140)

plt.title("Marks Obtained in Different Subjects")

plt.show()
```

Slip no 16:

Q.2 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. [10]

Please refer from above program …Already Solved question

Q.2 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

data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve', 'Fiona', 'Gaurav', 'Hari', 'Isha', 'Jai'],
    'Graduation Percentage': [85, 92, 78, 88, 75, 90, 81, 87, 93, 79],
    'Age': [21, 23, 22, 24, 20, 22, 23, 21, 25, 22]
}
df = pd.DataFrame(data)
average_age = df['Age'].mean()
average_percentage = df['Graduation Percentage'].mean()
print("Student Information:")
print(df)
print(f"Average Age of Students: {average_age}")
print(f"Average Graduation Percentage: {average_percentage}")
```

Slip no 17:

Q.2 A) Write a Python program to draw scatter plots to compare two features of the iris dataset [10]

Please refer from above program …Already Solved question

Q.2 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.

Please refer from above program …Already Solved question

Slip no 18:

Q.2 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) [10]

Please refer from above program …Already Solved question

Q.2 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

Please refer from above program …Already Solved question

Slip no 19:

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 dataframe.
2. To print the shape, number of rows-columns, data types, feature names and the description of the data
3. To Add 5 rows with duplicate values and missing values. Add a column ‘remarks’ with empty values. Display the data.

```
import pandas as pd


data = {
    'name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve', 'Fiona', 'Gaurav', 'Hari', 'Isha', 'Jai'],
    'age': [23, 25, 22, 24, 20, 22, 23, 21, 25, 22],
    'percentage': [85, 92, 78, 88, 75, 90, 81, 87, 93, 79]
}

df = pd.DataFrame(data)
print(df)
# Print the shape (number of rows and columns)
shape = df.shape
print(f"Shape of the DataFrame: {shape}")

# Print the number of rows
num_rows = df.shape[0]
print(f"Number of Rows: {num_rows}")

# Print the number of columns
num_columns = df.shape[1]
print(f"Number of Columns: {num_columns}")

# Print data types, feature names, and description
data_types = df.dtypes
feature_names = df.columns
description = df.describe()

print("Data Types:")
print(data_types)

print("\nFeature Names:")
print(feature_names)

print("\nDescription of the Data:")
print(description)
new_data = {
    'name': ['Alice', 'Bob', 'Charlie', 'Alice', 'David'],
    'age': [23, 25, None, None, 24],
    'percentage': [85, 92, 78, None, None]
}
df = df.append(pd.DataFrame(new_data), ignore_index=True)


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


# Display the updated DataFrame
print(df)
```

_Slip-20

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 color, labels and styling options. [10]
Q.2 B) Add two outliers to the above data and display the box plot. [5]

Please refer from above program …Already Solved question

Slip 21:

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.
[5]

Please refer from above program …Already Solved question

Slip 22:

Q.2) Dataset Name: winequality-red.csv [15]
Write a program in python to perform following tasks
a. Rescaling: Normalised the dataset using MinMaxScaler class
b. Standardizing Data (transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1)
c. Normalizing Data (rescale each observation to a length of 1 (a unit norm). For this, use the Normalizer class.)

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import Normalizer
df = pd.read_csv(r'D:\DS_Final\exam\TYDATA\winequality-red.csv')
features = df.drop('quality', axis=1)
scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(features)
scaled_df = pd.DataFrame(scaled_data, columns=features.columns)
```

```
# Display the scaled DataFrame
print("Original Data:")
print(df.head())  # Display the original data
print("\nScaled Data:")
print(scaled_df.head())  # Display the scaled data
# Create a Normalizer object
normalizer = Normalizer()
# Transform the data using the Normalizer
normalized_data = normalizer.transform(features)
# Create a new DataFrame with the normalized data
normalized_df = pd.DataFrame(normalized_data, columns=features.columns)
# Display the normalized DataFrame
print("Original Data:")
print(df.head())  # Display the original data
print("\nNormalized Data:")
print(normalized_df.head())  # Display the normalized data
```

Slip 23:

Q.2) Dataset Name: winequality-red.csv [15]
Write a program in python to perform following task
a. Rescaling: Normalised the dataset using MinMaxScaler class
b. Standardizing 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)

Please refer from above program …Already Solved question

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. [5]

Please refer from above program …Already Solved question

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 color, 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.

Please refer from above program …Already Solved question

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 color, 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 bar chart.

Please refer from above program …Already Solved question

Slip 27:

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

```
import pandas as pd

from sklearn.preprocessing import LabelEncoder

data = {

  'country': ['USA', 'Canada', 'UK', 'Germany', 'France', 'Canada', 'USA', 'UK', 'France', 'Germany'],

  'purchased': ['Yes', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes']

}


df = pd.DataFrame(data)


# Save the DataFrame to a CSV file named 'data.csv'

df.to_csv(r'D:\DS_Final\exam\TYDATA\new_data.csv', index=False)

print(df)


# Apply one-hot encoding to the 'country' column

df_one_hot_coding = pd.get_dummies(df, columns=['country'], prefix=['country'])

print(df_one_hot_coding)


# Initialize the LabelEncoder

label_encoder = LabelEncoder()


# Apply label encoding to the 'purchased' column

df['purchased'] = label_encoder.fit_transform(df['purchased'])
```

```
print(df)
```

Slip no: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 dataframe.

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 no 29:

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

Please refer from above program …Already Solved question

Slip no 30:

Q.2) Write 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 color, 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 bar chart.

Please refer from above program …Already Solved question