SRM Institute of Science and Technology

Department of Computer Applications

Delhi – Meerut Road, Sikri Kalan, Ghaziabad, Uttar Pradesh – 201204

Circular – 2023-24

BCA DS 1st Sem

Fundamentals of Data Science (UDS23102J)

Lab Manual

Lab 1: Print a Statement

Aim

To write a Python script that prints a simple statement.

Procedure

1. Open a Python IDE or text editor.
2. Use the print() function to display a message.
3. Run the script.

Source Code

```
print("Welcome to Data Science Lab!")
```

Input

No input required.

Expected Output

Welcome to Data Science Lab!

Lab 2: Perform Analysis on a Simple Dataset

Aim

To perform basic analysis on a simple dataset for data science and business intelligence applications.

Procedure

1. Load the dataset using pandas.
2. Display its head and basic statistics.
3. Use functions like describe() and info() to understand the data.

Source Code

```
import pandas as pd

df = pd.read_csv('sample_dataset.csv')
print(df.head())
print(df.describe())
print(df.info())
```

Input

CSV file: sample_dataset.csv

Expected Output

Displays the first few rows, statistical summary, and dataset structure.

Lab 3: Swap Two Numbers

Aim

To write a Python program that swaps two numbers.

Procedure

1. Input two numbers.
2. Use tuple unpacking to swap values.
3. Print before and after swapping.

Source Code

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))

print("Before swapping: a =", a, "b =", b)

a, b = b, a

print("After swapping: a =", a, "b =", b)
```

Input

```
Enter first number: 10
Enter second number: 20
```

Expected Output

```
Before swapping: a = 10 b = 20
After swapping: a = 20 b = 10
```

Lab 4: Subset and Aggregate on Iris Dataset

Aim

To write a Python script to find subset of dataset by using subset(), aggregate() functions on the Iris dataset.

Procedure

1. Import the necessary libraries (pandas, seaborn).
2. Load the Iris dataset.
3. Use conditional filtering for subsets.
4. Use groupby() and agg() for aggregation.

Source Code

```
import seaborn as sns
import pandas as pd

iris = sns.load_dataset("iris")
subset_setosa = iris[iris['species'] == 'setosa']
print("Subset where species is setosa:\n", subset_setosa.head())
agg_result = iris.groupby('species').agg('mean')
print("\nAggregated Mean Values by Species:\n", agg_result)
```

Input

Built-in iris dataset from seaborn.

Expected Output

Subset of rows where species = 'setosa' and aggregated mean values by species.

Lab 5: Reading and Writing Data Files

Aim

To read different types of datasets (.txt, .csv) from web and disk and write them to a specific disk location.

Procedure

1. Use pandas to read .csv or .txt files.
2. Load files from a local path or URL.
3. Write the data to a specific location using to_csv().

Source Code

```
import pandas as pd

url = "https://people.sc.fsu.edu/~jburkardt/data/csv/hw_200.csv"
data_web = pd.read_csv(url)
print("Data from web:\n", data_web.head())
data_web.to_csv("downloaded_data.csv", index=False)
print("File written to 'downloaded_data.csv'")
```

Input

URL to CSV file.

Expected Output

Preview of dataset from the URL and confirmation: 'File written to downloaded_data.csv'.

Lab 6: Basic Python Functions

Aim

To install Python and apply basic Python functions.

Procedure

1. Install Python from python.org.
2. Practice using basic functions like print(), type(), len(), int().

Source Code

```
name = "Data Science"
print("Welcome to", name)
print("Length of name:", len(name))
print("Type of name:", type(name))
```

Input

No input required.

Expected Output

```
Welcome to Data Science
Length of name: 12
Type of name: <class 'str'>
```

Lab 7: Numerical Array Processing with NumPy

Aim

To install and perform numerical array processing using NumPy.

Procedure

1. Install NumPy using pip.
2. Create arrays and perform basic operations like mean, reshape.

Source Code

```
import numpy as np

arr = np.array([10, 20, 30, 40, 50])
print("Array:", arr)
print("Mean:", np.mean(arr))
print("Reshaped Array (5x1):\n", arr.reshape(5, 1))
```

Input

Array elements initialized in code.

Expected Output

Mean and reshaped array printed.

Lab 8: Descriptive Statistics on mtcars & cars

Aim

To find basic descriptive statistics using summary, str, and quartile functions on mtcars & cars datasets.

Procedure

1. Import dataset.
2. Use describe(), info(), and quantile() to analyze.

Source Code

```
import pandas as pd
import seaborn as sns

mtcars = sns.load_dataset('mpg')
print("Summary:\n", mtcars.describe())
print("\nStructure:\n", mtcars.info())
print("\nQuartiles:\n", mtcars.quantile([0.25, 0.5, 0.75]))
```

Input

Built-in mpg dataset used as a substitute for mtcars.

Expected Output

Displays summary statistics, data structure, and quartiles.

Lab 9: Correlation Matrix

Aim

To find the correlation matrix of a dataset.

Procedure

1. Load a dataset.
2. Use the corr() function to compute the correlation matrix.

Source Code

```
import pandas as pd
import seaborn as sns

iris = sns.load_dataset("iris")
correlation_matrix = iris.corr(numeric_only=True)
print("Correlation Matrix:\n", correlation_matrix)
```

Input

Iris dataset.

Expected Output

Correlation matrix printed.

Lab 10: Correlation Plot on Iris Dataset

Aim

To plot and visualize the correlation among data on the iris dataset.

Procedure

1. Compute correlation matrix.
2. Use heatmap to visualize.

Source Code

```
import seaborn as sns
import matplotlib.pyplot as plt

iris = sns.load_dataset("iris")
corr = iris.corr(numeric_only=True)
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title("Correlation Matrix of Iris Dataset")
plt.show()
```

Input

Iris dataset.

Expected Output

Correlation heatmap displayed.

Lab 11: Exploratory Data Analysis Using Pandas

Aim

To install and perform simple Exploratory Data Analysis (EDA) using Pandas.

Procedure

1. Load a dataset.
2. Perform basic EDA using head(), describe(), value_counts(), isnull().sum().

Source Code

```
import pandas as pd
import seaborn as sns

df = sns.load_dataset("titanic")
print(df.head())
print("\nData Info:\n")
print(df.info())
print("\nMissing Values:\n", df.isnull().sum())
print("\nSummary Statistics:\n", df.describe())
```

Input

Titanic dataset.

Expected Output

Displays data preview, info, null counts, and summary statistics.

Lab 12: Learn and Explore a Sample Dataset with Pandas

Aim

To install, import Pandas, and explore a sample dataset with it.

Procedure

1. Install Pandas using pip install pandas.
2. Import Pandas and load a dataset.
3. Use Pandas functions to understand the dataset.

Source Code

```
import pandas as pd

data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, 30, 35],
    'Score': [85, 90, 95]
}

df = pd.DataFrame(data)
print("Dataset:\n", df)
print("\nData Description:\n", df.describe())
```

Input

Defined inline dictionary.

Expected Output

Displays the DataFrame and basic statistics.

Lab 13: Explore Iris Dataset with Scikit-learn and Pandas

Aim

To install and import Scikit-learn and explore the Iris dataset with Pandas for ML modeling.

Procedure

1. Load dataset using sklearn.datasets.
2. Convert it to a Pandas DataFrame.
3. Analyze features and target.

Source Code

```
from sklearn.datasets import load_iris
import pandas as pd

iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target
print(df.head())
print("\nTarget Value Counts:\n", df['target'].value_counts())
```

Input

Scikit-learn Iris dataset.

Expected Output

Displays the Iris dataset and class distribution.

Lab 14: Explore Data Visualization with Matplotlib

Aim

To install and import Matplotlib and explore various data visualization graphs.

Procedure

1. Install Matplotlib.
2. Create sample plots: line, scatter, bar, pie.

Source Code

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4]
y = [10, 20, 25, 30]
plt.plot(x, y)
plt.title("Line Plot")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.show()
```

Input

Inline data.

Expected Output

Displays a simple line chart.

Lab 15: Find Outliers Using Plot

Aim

To find outliers in data using visual plots.

Procedure

1. Load dataset.
2. Use box plot to visually identify outliers.

Source Code

```
import seaborn as sns
import matplotlib.pyplot as plt

df = sns.load_dataset("iris")
sns.boxplot(x=df['sepal_length'])
plt.title("Boxplot of Sepal Length")
plt.show()
```

Input

Iris dataset.

Expected Output

Boxplot showing potential outliers in sepal_length.

Lab 16: Data Distributions Using Box and Scatter Plot

Aim

To find data distributions using box and scatter plots.

Procedure

1. Load dataset.
2. Use seaborn to create box and scatter plots.

Source Code

```
import seaborn as sns
import matplotlib.pyplot as plt

iris = sns.load_dataset("iris")
sns.boxplot(x='species', y='sepal_length', data=iris)
plt.title("Boxplot: Sepal Length by Species")
plt.show()
sns.scatterplot(x='sepal_length', y='petal_length', hue='species', data=iris)
plt.title("Scatter Plot of Sepal vs Petal Length")
plt.show()
```

Input

Iris dataset.

Expected Output

Boxplot and scatter plot visualizing distributions and relationships.

Lab 17: Histogram, Bar Chart, and Pie Chart

Aim

To plot histogram, bar chart, and pie chart on sample data.

Procedure

1. Create sample data.
2. Use Matplotlib to plot histogram, bar, and pie charts.

Source Code

```
import matplotlib.pyplot as plt

data = [20, 20, 30, 40, 50, 60, 60, 70]
plt.hist(data, bins=5)
plt.title("Histogram")
plt.show()
names = ['A', 'B', 'C']
values = [10, 20, 30]
plt.bar(names, values)
plt.title("Bar Chart")
plt.show()
labels = ['Python', 'Java', 'C++']
sizes = [45, 30, 25]
plt.pie(sizes, labels=labels, autopct='%1.1f%%')
plt.title("Pie Chart")
plt.show()
```

Input

Inline sample data.

Expected Output

Histogram, bar chart, and pie chart displayed.

Lab 18: Explore All Data Visualization Graphs (Matplotlib)

Aim

To install, import Matplotlib and explore a variety of data visualization graphs.

Procedure

1. Load sample data.
2. Demonstrate multiple plots using Matplotlib: line, bar, scatter, histogram, pie.

Source Code

```
Refer to Lab 14 and Lab 17 combined examples for all basic visualizations.
```

Input

Sample data used for visualizations.

Expected Output

Displays all common plot types.

Lab 19: Line Chart in Python

Aim

To create a line chart in Python using sample data.

Procedure

1. Create x and y axis values.
2. Use plt.plot() to create line chart.

Source Code

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]
plt.plot(x, y, marker='o')
plt.title("Prime Number Line Chart")
plt.xlabel("Index")
plt.ylabel("Prime Number")
plt.grid(True)
plt.show()
```

Input

X and Y values defined in script.

Expected Output

Line chart connecting prime number values.

Lab 20: Customizing a Plot

Aim

To customize a Python plot using Matplotlib.

Procedure

1. Create a basic plot.
2. Customize title, axes, style, color, and markers.

Source Code

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4]
y = [2, 4, 1, 3]
plt.plot(x, y, color='green', marker='s', linestyle='--', linewidth=2)
plt.title("Customized Plot")
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
plt.grid(True)
plt.show()
```

Input

Inline data with custom styling.

Expected Output

Customized line plot with grid, color, markers, and style.