# Lab Experiments

**ROLL NO:231501133**

**Exp-1** Setting up the Python environment and libraries-Juypter Notebook PROGRAM:

print(2+3)

-Lists

* \*\*Bold text\*\*
* \*Italic text\*
* [Links](https://jupyter.org/)

import ipywidgets as widgets

from IPython.display import display

slider = widgets.IntSlider(value=25, min=0, max=100, description='Slider:') display(slider)

OUTPUT: 5

# -Lists Bold text Italic text Links

****

EXP-2 Data Import and Export

PROGRAM:

import pandas as pd

# Replace with your CSV file URL

url = 'https://raw.githubusercontent.com/kwaldenphd/eda pandas/main/data/titanic.csv'

df\_csv = pd.read\_csv(url)

# Display the first few rows df\_csv.head()

df\_excel = pd.read\_excel("/content/output.xlsx") # Replace with uploaded file name

print("Excel Data:") print(df\_excel.head())

from google.colab import drive drive.mount('/content/drive')

# Create sample SQLite database and table (for demo) engine = create\_engine('sqlite://', echo=False) df\_sample = pd.DataFrame({

"Name": ["Alice", "Bob", "Charlie"], "Age": [25, 30, 35]

})

df\_sample.to\_sql("people", con=engine, index=False)

# Read from the SQL table

df\_sql = pd.read\_sql("SELECT \* FROM people", engine) print("SQL Data:")

print(df\_sql)

# Read HTML table from a webpage url =

"https://en.wikipedia.org/wiki/List\_of\_countries\_by\_GDP\_(nominal)" tables = pd.read\_html(url)

# Display the first table df\_web = tables[0] print("Web Table Data:")

print(df\_web.head()) import pandas as pd

# Sample DataFrame

data = {'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 35],

'City': ['New York', 'San Francisco', 'Los Angeles']} df = pd.DataFrame(data)

# Export to Excel df.to\_excel('output1.xlsx', index=False)

OUTPUT:

# PassengerId Survived Pclass Name Sex Age SibSp Parch Braund,

**0** 1 0 3 **1** 2 1 1 **2** 3 1 3 **3**

4 1 1 **4** 5 0 3

Mr. Owen Harris

Cumings, Mrs. John Bradley

(Florence Briggs Th...

Excel Data:

PassengerId Survived Pclass \

0 1 0 3

1 2 1 1

2 3 1 3

3 4 1 1

4 5 0 3

SQL Data:

Heikkinen, Miss. Laina

Futrelle, Mrs. Jacques Heath

(Lily May Peel)

Allen, Mr. William Henry

male 22.0 1 female 38.0

1 female 26.0 0 female

35.0 1 male 35.0 0

Name Age

0 Alice 25

1 Bob 30

2 Charlie 35

Web Table Data:

0

0 Largest economies in the world by GDP (nominal...

\

EXP-3 Data Cleaning

PROGRAM:

import pandas as pd

titanic = pd.read\_csv('/content/drive/MyDrive/Colab Notebooks/FDVA\_CSV/Titanic/train (1).csv')

titanic.head() titanic.isnull().sum() import pandas as pd

from sklearn.preprocessing import StandardScaler, MinMaxScaler

titanic['Age'] = titanic['Age'].fillna(titanic['Age'].median())

titanic['Embarked'] =

titanic['Embarked'].fillna(titanic['Embarked'].mode()[0])

if 'Cabin' in titanic.columns:

titanic = titanic.drop('Cabin', axis=1)

print("\nMissing values after handling:\n", titanic.isnull().sum())

duplicates = titanic.duplicated().sum() print(f"\nNumber of duplicate rows: {duplicates}")

titanic.drop\_duplicates(inplace=True)

print("\nData types before conversion:\n", titanic.dtypes)

cat\_cols = ['Survived', 'Pclass', 'SibSp', 'Parch'] for col in cat\_cols:

titanic[col] = titanic[col].astype('category')

titanic['Fare'] = titanic['Fare'].astype(float) titanic['Age'] = titanic['Age'].astype(float)

print("\nData types after conversion:\n", titanic.dtypes)

num\_cols = ['Age', 'Fare']

scaler = StandardScaler()

titanic[num\_cols] = scaler.fit\_transform(titanic[num\_cols])

print("\nData after normalization:\n", titanic.head())

titanic.to\_csv('titanic\_cleaned.csv', index=False)

OUTPUT:

PassengerId Survived PclassName Sex Age SibSp Parch TicketFare Cabin Embarked

1. 1 0 3 Braund, Mr. Owen Harris male 22.0 1 0 A/5 21171 7.2500 NaN S
2. 2 1 1 Cumings, Mrs. John Bradley (Florence Briggs Th... female 38.0 1 0

PC 17599 71.2833 C85 C

1. 3 1 3 Heikkinen, Miss. Laina female 26.0 0 0 STON/O2. 3101282 7.9250 NaN S
2. 4 1 1 Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35.0 1 0

113803 53.1000 C123 S

1. 5 0 3 Allen, Mr. William Henry male 35.0 0 0 373450 8.0500 NaN S

**ket Cab Tic**

**Passen gerId**

**meSex**

**Survi ved Pcl ass**

**Ag Si Na**

**bS**

**e**

**p**

Braun

**Pa rc h**

**Fare**

**in** A/5

**Emba rked**

**0** 1 0 3 **1** 2 1 1

**2** 3 1 3

d, Mr. Owen Harris

Cumi ngs, Mrs.

John Bradl ey (Flore nce

**3** 4 1 1

Brigg s Th...

Heikki nen, Miss. Laina

Futrel le, Mrs.

Jacqu mal e

fem ale

fem ale

es

Heath (Lily May

Peel)

ale

fem 22.

NaN S

38.

01 0

01 0

C85 C

26.

53.1

C123 S

01 01138

000

00 0 35.

2117 1

PC 1759 9

STO

N/O2. 3101

282

7.25

03

00

71.2

833

7.92

50

NaN S

**4** 5 0 3 Allen, mal 35. 0 0 3734 8.05 NaN S

**meSex Ag**

**eSi**

**p**

**Pa rc h**

**ket Fare**

**Passen gerId**

e 0 50 00

**0**

**Survi ved Pcl ass**

**Na bS**

**Name** 0

**Sex** 0

**Age** 177

**Cab**

**Tic in**

**Emba rked**

**PassengerId** 0

**Survived** 0

**Pclass** 0

**SibSp** 0

**Parch** 0

**Ticket** 0

**Fare** 0

**Cabin** 687

**Embarked** 2

**dtype:** int64

Mr. Willia

Missing values after handling: PassengerId 0

Survived 0

Pclass 0

Name 0

Sex 0

Age 0

SibSp 0

Parch 0

Ticket 0

Fare 0

Embarked 0 dtype: int64

Number of duplicate rows: 0

Data types before conversion: PassengerId int64

Survived int64 Pclass int64 Name object Sex object Age float64 SibSp int64 Parch int64 Ticket object Fare float64

Embarked object dtype: object

Data types after conversion: PassengerId int64

Survived category Pclass category Name object

Sex object Age float64 SibSp category Parch category Ticket object Fare float64

Embarked object dtype: object

Data after normalization: PassengerId Survived Pclass \

0 1 0 3

1 2 1 1

2 3 1 3

3 4 1 1

4 5 0 3

Name Sex Age SibSp \ 0 Braund, Mr. Owen Harris male -0.565736 1 1 Cumings, Mrs. John Bradley (Florence Briggs Th... female 0.663861 1 2 Heikkinen, Miss. Laina female -0.258337 0

1. Futrelle, Mrs. Jacques Heath (Lily May Peel) female 0.433312 1 4 Allen, Mr. William Henry male 0.433312 0

Parch Ticket Fare Embarked

0 0 A/5 21171 -0.502445 S

1 0 PC 17599 0.786845 C

2 0 STON/O2. 3101282 -0.488854 S

3 0 113803 0.420730 S

1. 0 373450 -0.486337 S

EXP-4 -Data Inspection and Analysis

import pandas as pd import numpy as np

from sklearn.datasets import load\_iris

# Load the Iris dataset from sklearn iris = load\_iris()

df = pd.DataFrame(data=iris.data, columns=iris.feature\_names) # Add the species column df['species'] = pd.Categorical.from\_codes(iris.target,

iris.target\_names) df.head() # View first 5 rows df.tail() # View last 5 rows

df.info() # Summary: data types, nulls

df.describe() # Quick stats for numerical columns df.columns #colummn names

df.shape # Rows and columns count

df[df['species'] == 'setosa']

df[(df['species'] == 'setosa') & (df['sepal length (cm)'] >

5.0)] df[['sepal length (cm)', 'sepal width (cm)']] df['sepal length (cm)'].mean() # Mean

df['sepal length (cm)'].median() # Median

df['sepal length (cm)'].mode() # Mode (returns a Series)

df['sepal length (cm)'].min(), df['sepal length (cm)'].max() # Range df['sepal length (cm)'].var() # Variance

df['sepal length (cm)'].std() # Standard Deviation df.corr(numeric\_only=True)

OUTPUT:

RangeIndex: 150 entries, 0 to 149 Data columns (total 4 columns):

# Column Non-Null Count Dtype

1. sepal length (cm) 150 non-null float64
2. sepal width (cm) 150 non-null float64
3. petal length (cm) 150 non-null float64
4. petal width (cm) 150 non-null float64 dtypes: float64(4)

memory usage: 4.8 KB

(150, 4)

# sepal length (cm)

**sepal width (cm) petal length (cm) species petal width**

**0** 5.1 3.5 1.4 0.2 setosa **1** 4.9 3.0 1.4 0.2 setosa **2** 4.7 3.2 1.3 0.2 setosa sepal length (cm) sepal width (cm) petal length (cm) petal width

**(cm)**

(cm) species

0 5.1 3.5 1.4 0.2 setosa

5 5.4 3.9 1.7 0.4 setosa

10 5.4 3.7 1.5 0.2 setosa

14 5.8 4.0 1.2 0.2 setosa

15 5.7 4.4 1.5 0.4 setosa

# sepal length (cm) sepal width (cm)

**0** 5.1 3.5

**1** 4.9 3.0

**2** 4.7 3.2

**3** 4.6 3.1

**4** 5.0 3.6

**148** 6.2 3.4

**149** 5.9 3.0

Sepal length

**0** 5.0

**dtype:** float64

0.8280661279778629

sepal length (cm) sepal width (cm) petal length (cm) petal width (cm) sepal length (cm) 1.000000 -0.117570 0.871754 0.817941 sepal width

(cm) -0.117570 1.000000 -0.428440 -0.366126 petal length (cm)

0.871754 -0.428440 1.000000 0.962865 petal width (cm) 0.817941

-0.366126 0.962865 1.000000

Exp - 5 Data Visualization using Matplotlib