**FDVA 1-5 LAB EXPERIMENT**

**AD23632**

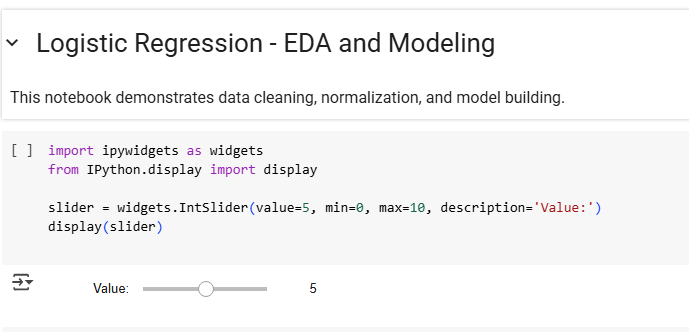
**Exp 1**

# Python code: A simple example

import pandas aspd

print("Welcome to AI & ML Jupyter Notebook!")

output:Welcome to AI & ML Jupyter Notebook!



**Exp 2**

#exp2

import pandas aspd

fromsqlalchemyimportcreate\_engine

import sqlite3

csv\_url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"

df\_csv = pd.read\_csv(csv\_url)

print("CSV Data (Titanic) sample:")

print(df\_csv.head())

df\_csv.head(10).to\_excel("sample\_titanic.xlsx", index=False)

df\_excel = pd.read\_excel("sample\_titanic.xlsx")

print("\nExcel Data sample:")

print(df\_excel.head())

conn = sqlite3.connect(':memory:')

df\_csv.to\_sql('titanic', conn, index=False, if\_exists='replace')

query = "SELECT \* FROM titanic LIMIT 5"

df\_sql = pd.read\_sql(query, conn)

print("\nSQL Data sample:")

print(df\_sql)

url = 'https://en.wikipedia.org/wiki/List\_of\_countries\_by\_GDP\_(nominal)'

tables = pd.read\_html(url)

print(f"\nNumber of tables scraped from webpage: {len(tables)}")

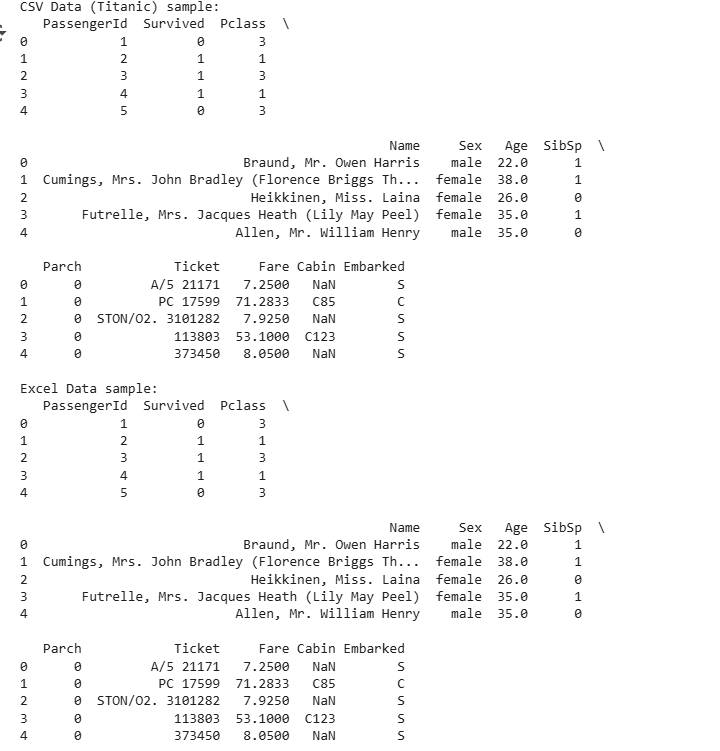
df\_web = tables[0]

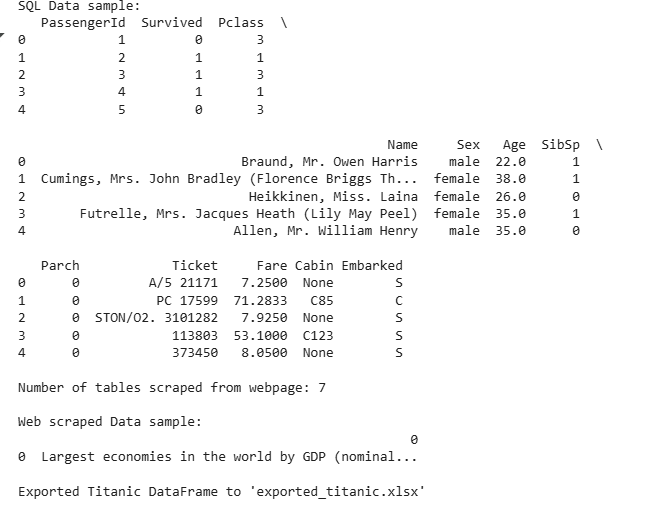
print("\nWeb scraped Data sample:")

print(df\_web.head())

df\_csv.to\_excel('exported\_titanic.xlsx', index=False)

print("\nExported Titanic DataFrame to 'exported\_titanic.xlsx'")

**output:**



**Exp 3:**

#exp3

import pandas aspd

importnumpyasnp

fromsklearn.preprocessingimportStandardScaler, MinMaxScaler

file\_path = '/content/logistic\_regression\_dataset.csv'

df = pd.read\_csv(file\_path)

print("Missing values per column:\n", df.isnull().sum())

for col indf.columns:

    ifdf[col].dtypein ['int64', 'float64']:

        df[col].fillna(df[col].mean(), inplace=True)

    else:

        df[col].fillna(df[col].mode()[0], inplace=True)

df.dropna(thresh=len(df.columns) - 1, inplace=True)

df.drop\_duplicates(inplace=True)

df = df.loc[:, ~df.columns.str.contains('^Unnamed'if'ID'indf.columns:

    df.drop(columns=['ID'], inplace=True)

for col indf.select\_dtypes(include='object').columns:

    ifdf[col].nunique() <50:

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

for col indf.columns:

    ifdf[col].dtype == 'object':

        try:

            df[col] = pd.to\_numeric(df[col])

        except:

            pass

numeric\_cols = df.select\_dtypes(include=['int64', 'float64']).columns

scaler\_std = StandardScaler()

df\_standardized = df.copy()

df\_standardized[numeric\_cols] = scaler\_std.fit\_transform(df[numeric\_cols])

scaler\_mm = MinMaxScaler()

df\_minmax = df.copy()

df\_minmax[numeric\_cols] = scaler\_mm.fit\_transform(df[numeric\_cols])

print("Cleaned Data Sample:")

print(df.head())

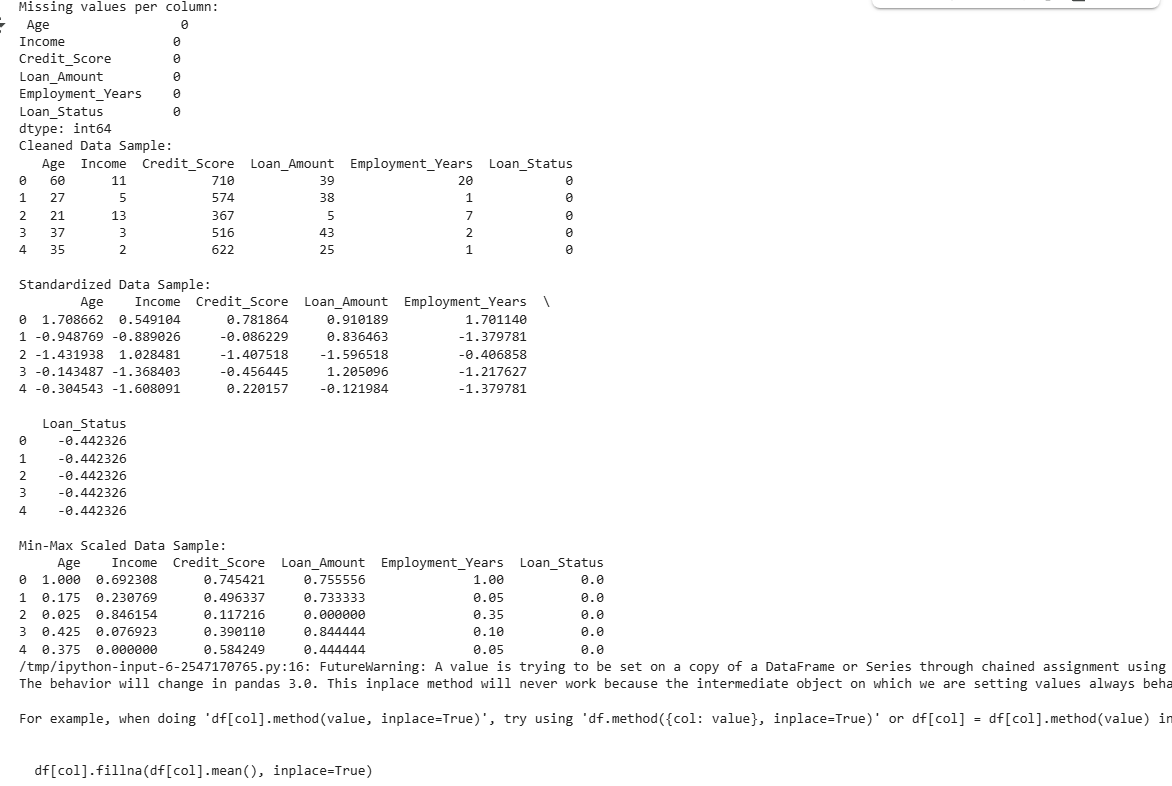
print("\nStandardized Data Sample:")

print(df\_standardized.head())

print("\nMin-Max Scaled Data Sample:")

print(df\_minmax.head())

**output:**



**Exp 4:**

#exp4

import pandas aspd

importseabornassns

importmatplotlib.pyplotasplt

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv'

df = pd.read\_csv(url, sep=';')

print("First 5 rows:")

print(df.head())

print("\nShape of dataset:", df.shape)

print("\nColumn names:")

print(df.columns)

print("\nInfo about data types and null values:")

print(df.info())

print("\nSummary statistics:")

print(df.describe())

print("\nRows where quality >= 7:")

print(df[df['quality'] >= 7].head())

print("\nSelect columns: alcohol and quality")

print(df[['alcohol', 'quality']].head())

print("\nDescriptive statistics for 'alcohol':")

print(f"Mean: {df['alcohol'].mean():.2f}")

print(f"Median: {df['alcohol'].median():.2f}")

print(f"Mode: {df['alcohol'].mode().values}")

print("\nDescriptive statistics for 'quality':")

print(f"Range: {df['quality'].max() - df['quality'].min()}")

print(f"Variance: {df['quality'].var():.2f}")

print(f"Standard Deviation: {df['quality'].std():.2f}")

plt.figure(figsize=(8,4))

sns.histplot(df['alcohol'], kde=True, bins=30)

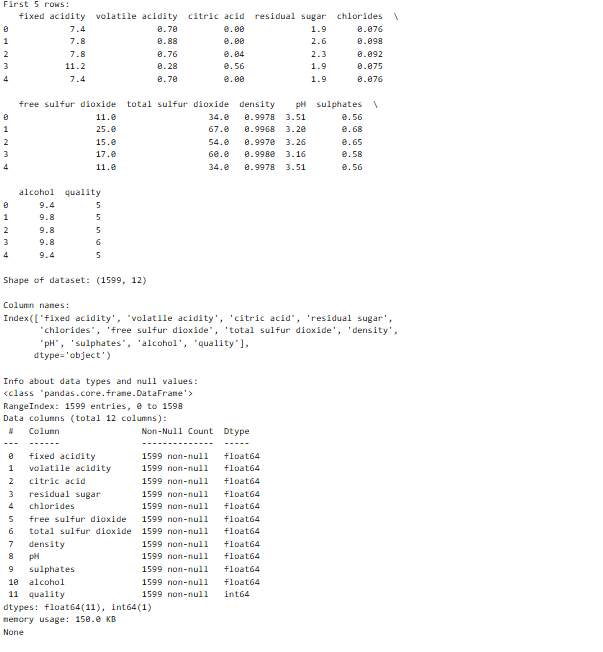
plt.title('Alcohol Content Distribution')

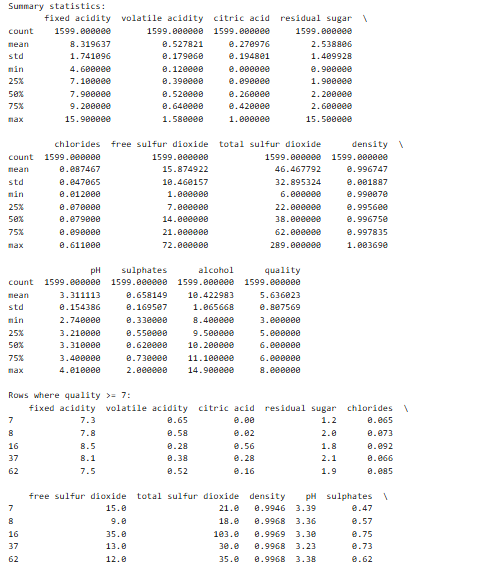
plt.xlabel('Alcohol')

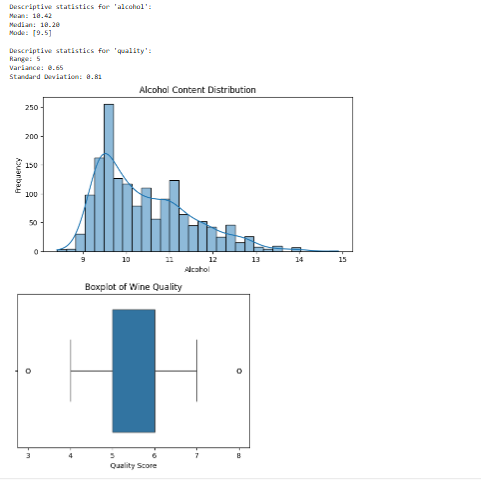
plt.ylabel('Frequency')

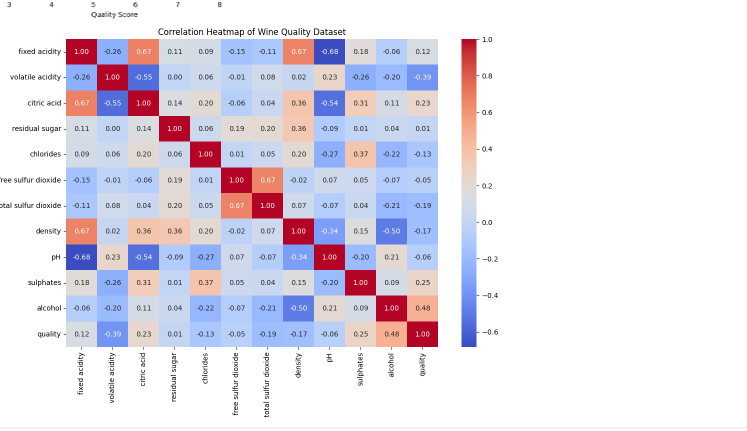
plt.show()

**output:**









Exp 5

#exp5

import pandas as pd

import matplotlib.pyplot as plt

# Load the dataset from the GitHub Gist

url = 'https://gist.githubusercontent.com/svlsml/3b5dd6723510ae43a0f121213f6583fc/raw/StudentsPerformance.csv'

df = pd.read\_csv(url)

# Preview data

print(df.head())

# Prepare data

x = df.index  # Use row index for line plot

y\_math = df['math\_score']

y\_reading = df['reading\_score']

# For bar chart: average scores by gender

avg\_by\_gender = df.groupby('gender')[['math\_score', 'reading\_score', 'writing\_score']].mean()

# For histogram: distribution of math scores

hist\_data = df['math\_score']

# Generate subplots

fig, axs = plt.subplots(1, 3, figsize=(18, 5))

# 1. Line chart: Math vs Reading

axs[0].plot(x, y\_math, label='Math Score', alpha=0.7)

axs[0].plot(x, y\_reading, label='Reading Score', alpha=0.7)

axs[0].set\_title('Line Chart: Math vs Reading Scores')

axs[0].set\_xlabel('Sample Index')

axs[0].set\_ylabel('Score')

axs[0].legend()

axs[0].grid(True)

# 2. Bar chart: Average by gender

avg\_by\_gender.plot(kind='bar', ax=axs[1])

axs[1].set\_title('Average Scores by Gender')

axs[1].set\_xlabel('Gender')

axs[1].set\_ylabel('Average Score')

axs[1].legend(title='Subject')

axs[1].grid(axis='y')

# 3. Histogram: Math score distribution

axs[2].hist(hist\_data, bins=20, color='purple', edgecolor='black')

axs[2].set\_title('Histogram: Math Score Distribution')

axs[2].set\_xlabel('Math Score')

axs[2].set\_ylabel('Frequency')

# Adjust layout and show

plt.tight\_layout()

plt.show()

output:

