#q1

# Solve for x, y & z where: [05 marks]

# Eq1 : x + y + z = 2 | Eq2 : 6x - 4y + 5z = 31 | Eq3: 5x + 2y + 2z = 13

# Coefficients of the equations

import numpy as np

coefficients = np.array([[1, 1, 1],

[6, -4, 5],

[5, 2, 2]])

# Right-hand side of the equations

rhs = np.array([2, 31, 13])

# Solve the system of equations

solution = np.linalg.solve(coefficients, rhs)

# Extract the values of x, y, and z from the solution

x = solution[0]

y = solution[1]

z = solution[2]

# Print the solution

print("Solution:")

print(f"x = {x}")

print(f"y = {y}")

print(f"z = {z}")

# 2.1

#creating a dictionary

emp\_dict = {

'Name': ['ABC', 'DEF', 'GHI', 'JKL', 'MNO', 'PQR', 'STU'],

'Education': ['Graduate', 'Postgraduate', 'Postgraduate', 'Graduate', 'Graduate', 'Postgraduate', 'Graduate'],

'Gender': ['Male', 'Female', 'Male', 'Other', 'Female', 'Female', 'Male'],

}

emp\_dict

#2.2

emp\_dict['age']=[22,27,26,23,24,30,21]

emp\_dict

#3.1

import pandas as pd

emp\_df = pd.DataFrame(emp\_dict)

emp\_df

#3.2

salary\_lakhs = [6, 15, 20, 5, 10, 18, 12]

bonus\_percent = [12.50, 8.75, 6.25, 10.20, 13.60, 11.40, 9.80]

emp\_df['Salary\_Lakhs'] = salary\_lakhs

emp\_df['Bonus%'] = bonus\_percent

#3.3

new\_data = {

'Name': ['VWX', 'YZA', 'BCD'],

'Education': ['Postgraduate', 'Graduate', 'Postgraduate'],

'Gender': ['Male', 'Female', 'Other'],

'age': [35, 28, 32],

'Salary\_Lakhs': [14, 7, 8],

'Bonus%': [5.50, 7.75, 14.80]

}

new\_row = pd.DataFrame(new\_data)

# Append the new row to the original DataFrame

emp\_df = emp\_df.append(new\_row, ignore\_index=True)

emp\_df

#3.4

emp\_df['Gross\_Salary\_Lakhs'] = emp\_df['Salary\_Lakhs'] \* (1 + emp\_df['Bonus%'])

# Print the updated 'emp\_df'

print(emp\_df)

#4.1

# Create a Subset ‘emp\_df\_ss’ from ‘emp\_df’ with the following Variables: {Name, Age, Gross\_Salary\_Lakhs}

# List of columns to include in the subset

selected\_columns = ['Name', 'age', 'Gross\_Salary\_Lakhs']

# Create the subset DataFrame emp\_df\_ss

emp\_df\_ss = emp\_df[selected\_columns]

emp\_df\_ss

# 5.1 Create a Copy of 'emp\_df' Named as 'emp\_df\_age\_sorted' and Sort {Highest to Lowest} by 'Age'

emp\_df\_age\_sorted = emp\_df.copy()

# Sort emp\_df\_age\_sorted by 'Age' in descending order

emp\_df\_age\_sorted=emp\_df\_age\_sorted.sort\_values(by='age', ascending=False)

emp\_df\_age\_sorted

# 5.2 Create a Copy of 'emp\_df' Named as 'emp\_df\_age\_salary\_sorted' and Sort: First by 'Age' {Lowest to Highest}, Second by 'Gross\_Salary\_Lakhs' {Highest to Lowest}

# Create a copy of emp\_df

emp\_df\_age\_salary\_sorted = emp\_df.copy()

# Sort emp\_df\_age\_salary\_sorted first by 'Age' (ascending), then by 'Gross\_Salary\_Lakhs' (descending)

emp\_df\_age\_salary\_sorted = emp\_df\_age\_salary\_sorted.sort\_values(by=['age', 'Gross\_Salary\_Lakhs'], ascending=[True, False])

emp\_df\_age\_salary\_sorted

#6.1 Create a Dataframe ‘emp\_df\_filtered’ to Filter ‘emp\_df’ using the following Information: 'Age' >= 25 & 'Gender' = 'Female'

emp\_df\_filtered= emp\_df[(emp\_df['age'] >=25) & (emp\_df['Gender']=="Female")]

emp\_df\_filtered

#6.2 Create 2 Subsets: ‘emp\_df\_grad’ & ‘emp\_df\_postgrad’ from ‘emp\_df’ containing Information of Employees having ‘Education’ as ‘Graduate’ & ‘Postgraduate’, respectively

# Create 'emp\_df\_grad' containing employees with 'Education' as 'Graduate'

emp\_df\_grad = emp\_df[emp\_df['Education'] == 'Graduate']

# Create 'emp\_df\_postgrad' containing employees with 'Education' as 'Postgraduate'

emp\_df\_postgrad = emp\_df[emp\_df['Education'] == 'Postgraduate']

print(emp\_df\_grad)

print(emp\_df\_postgrad)

# 7.1. Create a Dataframe ‘emp\_df\_merged’ to Inner Merge ‘emp\_df\_grad’ having only following Variables {Name, Gender, Age} with ‘emp\_df\_postgrad’ having only following Variables {Name, Gender, Gross\_Salary\_Lakhs} on ‘Gender

# Select the columns for each DataFrame

# Assuming emp\_df\_grad and emp\_df\_postgrad are our DataFrames with different column names

columns\_grad = ['Name', 'Gender', 'age']

emp\_df\_grad = emp\_df[columns\_grad]

# Select columns for emp\_df\_postgrad

columns\_postgrad = ['Name', 'Gender', 'Gross\_Salary\_Lakhs']

emp\_df\_postgrad = emp\_df[columns\_postgrad]

# Perform an inner merge on 'Gender' to create emp\_df\_merged

emp\_df\_merged = emp\_df\_grad.merge(emp\_df\_postgrad, on='Gender', how='inner')

print(emp\_df\_merged)

#8.1 Group ‘emp\_df’ to Create a Table ‘emp\_df\_gen\_edu’ using ‘Gender’ & ‘Education’ having Count of Employees

# emp\_df is original DataFrame

# Group by 'Gender' and 'Education' and count employees in each group

emp\_df\_gen\_edu = emp\_df.groupby(['Gender', 'Education']).size().reset\_index(name='Count of employees')

emp\_df\_gen\_edu

# 8.2 Group ‘emp\_df’ to Create a Table ‘emp\_df\_gen\_age\_sal’ using ‘Gender’ with Average of Variables {Age & Gross\_Salary\_Lakhs}

# Assuming emp\_df is your original DataFrame

# Group by 'Gender' and calculate the average of 'Age' and 'Gross\_Salary\_Lakhs' in each group

emp\_df\_gen\_age\_sal = emp\_df.groupby('Gender')[['age', 'Gross\_Salary\_Lakhs']].mean().reset\_index()

emp\_df\_gen\_age\_sal

#9.1 Create Panel Dataframe from Cross-Sectional Dataframe

data = {

'Company': ['ZYX', 'WVU', 'TSR'],

2023: [123, 456, 789],

2024: [321, 654, 987]

}

df\_cross\_section = pd.DataFrame(data)

# Melt the Cross-Sectional DataFrame to transform it into the Panel DataFrame format

df\_panel = pd.melt(df\_cross\_section, id\_vars='Company', var\_name='Year', value\_name='Profit')

# Display the Panel DataFrame 'df\_panel'

print("Panel DataFrame:")

print(df\_panel)