import pandas as pd  
  
  
def readFile(path):  
 pd.set\_option('display.max\_columns', None)  
 pd.set\_option('display.max\_rows', None)  
 pd.set\_option('display.width', 1000)  
 pd.options.mode.chained\_assignment = None  
 file = pd.read\_csv(path)  
 return file  
  
  
*# 1. load the ”Sleep\_Efficiency.csv” file into Pandas*def Task1():  
 df = readFile('Sleep\_Efficiency.csv')  
 print("Task 1:")  
 print("Finished reading files.")  
 print("Origin Data Sample:\n")  
 print("Sleep Efficiency:")  
 print(df.head())  
 print("\n", "\*" \* 50)  
  
  
*# 2. how many different entries are there for each columns.*def Task2():  
 df = readFile('Sleep\_Efficiency.csv')  
 print("Task 2:")  
 print("Different entries in each columns.")  
 print(df.nunique())  
 print("\n", "\*" \* 50)  
  
  
*# 3. how many entries are missing for each column*def Task3():  
 df = readFile('Sleep\_Efficiency.csv')  
 print("Task 3:")  
 print("Total missing entries in each columns.")  
 print(df.isna().sum())  
 print("\n", "\*" \* 50)  
  
  
*# 4. for each missing entry, impute the missing values.*def Task4():  
 df = readFile('Sleep\_Efficiency.csv')  
  
 for i in range(10, 15):  
 mylist = df[df.iloc[:, i].isnull()].index.tolist()  
 for j in range(len(mylist)):  
 age = df.iloc[mylist[j]]['Age']  
 gender = df.iloc[mylist[j]]['Gender']  
  
 *# (a) find 5 peoples with the same gender with the closest age (”neighbors”)* filter = df.loc[(df['Age']-age).abs().argsort()]  
 neignbor = filter.loc[(df['Gender'].str.contains(gender))]  
 neignbor = neignbor.head()  
  
 *# (b) compute the average of these ”neighbors” as the value* avg = neignbor.iloc[:, i].mean()  
 df.iloc[mylist[j], i] = avg  
  
 print("Task 4:")  
 print("Empty data has been filled. The following are sample data.")  
 print(df.head())  
 print("\nTotal missing data after filling:")  
 print(df.isna().sum())  
 print("\n", "\*" \* 50)  
 df.to\_csv('Sleep\_Efficiency\_filled.csv')  
  
  
*# 5. divide all people into the following groups:  
# • Group 1: children (1-12) • Group 2: teenagers (13-17) • Group 3: young adults (18-30)  
# • Group 4: adults (31-60) • Group 5: older adults (65+)  
# For each group, compute the mean and standard deviation and summarize them in three tables (like one below):  
# one table for females, one tables for males and one combined*def Task5():  
 df = readFile('Sleep\_Efficiency\_filled.csv')  
 df = df.drop('Unnamed: 0', axis=1)  
 group = [0] \* 6  
 print("Task 5:")  
  
 for i in range(3):  
 metric = {'Metric': ['Age(mean)', 'Age(STD)', 'Duration(mean)', 'Duration(STD)', 'Efficiency(mean)', 'Efficiency(STD)',  
 'REM %(mean)', 'REM %(STD)', 'Deep sleep %(mean)', 'Deep sleep %(STD)', 'Light sleep%(mean)', 'Light sleep%(STD)',  
 '# Awake(mean)', '# Awake(STD)', 'Smoking(mean)', 'Smoking(STD)', 'Exercise(mean)', 'Exercise(STD)']}  
 data = pd.DataFrame(metric)  
  
 if i == 0:  
 gender = 'Female'  
 print('Group by female: \n')  
 elif i == 1:  
 gender = 'Male'  
 print('Group by male:')  
 print('\*There has no data for male group 1 and 2.\n')  
 else:  
 gender = ''  
 print('Group by both genders: \n')  
  
 group[1] = filterData(df, gender, 1, 12) *# No male in this group* group[2] = filterData(df, gender, 13, 17) *# No male in this group* group[3] = filterData(df, gender, 18, 30)  
 group[4] = filterData(df, gender, 31, 60)  
 group[5] = filterData(df, gender, 65, 999)  
  
 for j in range(1,6):  
 avg\_age = group[j]['Age'].mean()  
 std\_age = group[j]['Age'].std()  
 avg\_dur = group[j]['Sleep duration'].mean()  
 std\_dur = group[j]['Sleep duration'].std()  
 avg\_eff = group[j]['Sleep efficiency'].mean()  
 std\_eff = group[j]['Sleep efficiency'].std()  
 avg\_rem = group[j]['REM sleep percentage'].mean()  
 std\_rem = group[j]['REM sleep percentage'].std()  
 avg\_deep = group[j]['Deep sleep percentage'].mean()  
 std\_deep = group[j]['Deep sleep percentage'].std()  
 avg\_light = group[j]['Light sleep percentage'].mean()  
 std\_light = group[j]['Light sleep percentage'].std()  
 avg\_awk = group[j]['Awakenings'].mean()  
 std\_awk = group[j]['Awakenings'].std()  
 avg\_exr = group[j]['Exercise frequency'].mean()  
 std\_exr = group[j]['Exercise frequency'].std()  
  
 writeData = [avg\_age, std\_age, avg\_dur, std\_dur, avg\_eff, std\_eff, avg\_rem, std\_rem,  
 avg\_deep, std\_deep, avg\_light, std\_light, avg\_awk, std\_awk,  
 'Yes/No', 'nan', avg\_exr, std\_exr]  
 groupNum = gender + ' Group ' + str(j)  
 data[groupNum] = writeData  
  
 print(data)  
 print("\*" \* 50, '\n')  
 if i == 0:  
 store = data  
 else:  
 data = data.drop([data.columns[0]], axis=1)  
 store = pd.concat([store, data], axis=1)  
 print("\n", "\*" \* 50)  
 return store  
  
  
def filterData(df, key, min, max):  
 temp = df.loc[(df['Age'] >= min) & (df['Age'] <= max)]  
 res = temp.loc[df['Gender'].str.contains(key)]  
 return res  
  
  
*# 6. Which group (age range and gender) sleeps the most? the least? wakes up the most? the least?*def Task6\_9():  
 df = Task5()  
 df1 = df.drop(['Male Group 1', 'Male Group 2', df.columns[0]], axis=1)  
 df1 = df1.drop([14, 15])  
 df1 = df1.astype(float)  
 df1['Max'] = df1.idxmax(axis=1, numeric\_only=True)  
 df1['Min'] = df1.idxmin(axis=1, numeric\_only=True)  
 df['Max'] = df1['Max']  
 df['Min'] = df1['Min']  
 pd.set\_option('display.max\_columns', 6)  
  
 print("Task 6:\n")  
 print(df)  
 print("\nWhich group (age range and gender) sleeps the most? the least? wakes up the most? the least?")  
 print("Based on the table, female group 1 sleeps the most; male group 4 sleeps the least.")  
 print("Male group 5 wakes up the most; female group 5 wakes up the least.")  
  
 *# 7. which group has the max and min sleep efficiency? Deep/- light sleep percentage* print("\nTask 7:")  
 print("which group has the max and min sleep efficiency? Deep/- light sleep percentage?")  
 print("Male group 4 has the max and Female group 1 has the min sleep efficiency.")  
 print("Male group 4 has max deep sleep percentage and female group 1 has the min.")  
 print("Female group 2 has the max light sleep percentage and Male group 4 has the min.")  
  
 *# 8. do people sleep more or less if they exercise?* print("\nTask 8:")  
 print("do people sleep more or less if they exercise?")  
 print("Male group 5 exercise the most but they are not the one sleep the most.")  
 print("Femlae group 1 exercise the lease but they are the one sleep the most.")  
  
 *# 9. do smokers sleep more or less* print("\nTask 9:")  
 print("Data not clear.")  
  
 print("\nTask 10")  
 print(df)  
  
  
  
def Driver():  
 Task1()  
 Task2()  
 Task3()  
 Task4()  
 Task6\_9()  
 *# 10. examine your tables carefully and tell us what you see (2-3 paragraphs)* print("\nI can see that Male group 5 is on the max list very often, and Female group 1 is often on the min side of the table.")  
 print("Group 4 is the only group that contains both genders that is on the max/min list, and it is related to the age.\n ")  
 print("In one way, exercise does help to improve the sleep, as you can see Male group, the one exercise the most, is the one \n"  
 "that shows on the max list the most. But we cannot say that exercise will definitely help one to sleep, as you can see\n"  
 "Female group 1 does not exersice at all but still has the longest sleeping duration. ")  
  
  
Driver()