Q2

April 15, 2024

[ ]:
```
import pandas as pd
import numpy as np
Quiz_data = pd.read_csv('Quiz.csv')
Homework_data = pd.read_csv('Homework.csv')
Exam_data = pd.read_csv('Exam.csv')
```

[ ]:
```
final_data = pd.merge(Quiz_data, Homework_data, on=['Roll␣
 ↪No','Name'],how="outer")
final_data=pd.merge(final_data, Exam_data,on=["Roll No","Name"],how="outer")
final_data
```

[ ]:
```
   Roll No      Name  Quiz1  Max Score_x  Quiz2  Max Score.1_x   HW1  \
0      101      Ajay      8           10     12             15   NaN
1      102      Rita      9           10     13             15  12.0
2      102  Shalmali     10           10     14             15  12.0

   Max Score_y  HW2  Max Score.1_y  Test1 Test2
0           15    9             10     23   Nan
1           15  Nan             10     22    22
2           15    6             10     21    21
```

[ ]:
```
final_data.fillna(0, inplace = True)
final_data
```

[ ]:
```
   Roll No      Name  Quiz1  Max Score_x  Quiz2  Max Score.1_x   HW1  \
0      101      Ajay      8           10     12             15   0.0
1      102      Rita      9           10     13             15  12.0
2      102  Shalmali     10           10     14             15  12.0

   Max Score_y  HW2  Max Score.1_y  Test1 Test2
0           15    9             10     23   Nan
1           15  Nan             10     22    22
2           15    6             10     21    21
```

[ ]:
```
final_data["Sum of Quiz Scores"] = final_data["Quiz1"] + final_data["Quiz2"]
final_data["Sum of Max Scores"] = final_data["Max Score_x"] + final_data["Max␣
 ↪Score.1_x"]
```

```
new_quiz = final_data[['Roll No', 'Name', 'Sum of Quiz Scores', 'Sum of Max␣
 ↪Scores']]

new_quiz['Total Quiz Score'] = new_quiz['Sum of Quiz Scores'] / new_quiz['Sum␣
 ↪of Max Scores']

new_quiz
```

```
C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\3359108876.py:6:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_quiz['Total Quiz Score'] = new_quiz['Sum of Quiz Scores'] / new_quiz['Sum
of Max Scores']
```

[ ]:

	Roll No	Name	Sum of Quiz Scores	Sum of Max Scores	Total Quiz Score
0	101	Ajay	20	25	0.80
1	102	Rita	22	25	0.88
2	102	Shalmali	24	25	0.96

[ ]:
```
# Calculate final homework and final max homework scores
final_data["HW1"] = pd.to_numeric(final_data["HW1"], errors='coerce') # coz␣
 ↪directly its showing error
final_data["HW2"] = pd.to_numeric(final_data["HW2"], errors='coerce')
final_data['HW2'].fillna(0, inplace=True)#coz hw2 still hd Nan

# Calculate final homework and final max homework scores
final_data["Sum of Hw Scores"] = final_data["HW1"] + final_data["HW2"]
final_data["Sum of Maximum Scores"] = final_data["Max Score_y"] +␣
 ↪final_data["Max Score.1_y"]
new_homework= final_data[['Roll No', 'Name', 'Sum of Hw Scores','Sum of Maximum␣
 ↪Scores']]
new_homework['Total Homework Score'] = (new_homework['Sum of Hw Scores'] /␣
 ↪new_homework['Sum of Maximum Scores'])
new_homework
```

```
C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\3477659495.py:4:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
```

```
instead, to perform the operation inplace on the original object.


  final_data['HW2'].fillna(0, inplace=True)#coz hw2 still hd Nan
C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\3477659495.py:10:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_homework['Total Homework Score'] = (new_homework['Sum of Hw Scores'] /
new_homework['Sum of Maximum Scores'])
```

[ ]:

	Roll No	Name	Sum of Hw Scores	Sum of Maximum Scores	\
0	101	Ajay	9.0	25	
1	102	Rita	12.0	25	
2	102	Shalmali	18.0	25	

	Total Homework Score
0	0.36
1	0.48
2	0.72

[ ]:
```
final_data["Test1"] = pd.to_numeric(final_data["Test1"], errors='coerce') # coz␣
 ↪directly its showing error
final_data["Test2"] = pd.to_numeric(final_data["Test2"], errors='coerce')
final_data['Test2'].fillna(0, inplace=True)#coz hw2 still hd Nan

# Calculate final homework and final max homework scores
final_data["Sum of Exam Score"] = final_data["Test1"] + final_data["Test2"]
final_data['Sum of Max Score'] = [50,50,50]
new_exam= final_data[['Roll No', 'Name', 'Sum of Exam Score','Sum of Max␣
 ↪Score']]
new_exam['Total Exam Score'] = (new_exam['Sum of Exam Score'] / new_exam['Sum␣
 ↪of Max Score'])
new_exam
```

```
C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\505153484.py:3:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.
```

```
  final_data['Test2'].fillna(0, inplace=True)#coz hw2 still hd Nan
C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\505153484.py:9:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_exam['Total Exam Score'] = (new_exam['Sum of Exam Score'] / new_exam['Sum
of Max Score'])
```

[ ]:

	Roll No	Name	Sum of Exam Score	Sum of Max Score	Total Exam Score
0	101	Ajay	23.0	50	0.46
1	102	Rita	44.0	50	0.88
2	102	Shalmali	42.0	50	0.84

[ ]:
```
# Assign weights to each category
weights = {'Total Quiz Score': 0.3, 'Total Homework Score': 0.3, 'Total Exam␣
 ↪Score': 0.4}

# Calculate the weighted scores for each category
new_quiz['Weighted_Quiz'] = new_quiz['Total Quiz Score'] * weights['Total Quiz␣
 ↪Score']
new_homework['Weighted_HW'] = new_homework['Total Homework Score'] *␣
 ↪weights['Total Homework Score']
new_exam['Weighted_Exam'] = new_exam['Total Exam Score'] * weights['Total Exam␣
 ↪Score']

# Concatenate the DataFrames along the columns axis (horizontally)
final = pd.concat([new_quiz['Weighted_Quiz'], new_homework['Weighted_HW'],␣
 ↪new_exam['Weighted_Exam']], axis=1)

# Calculate the final score by summing up the weighted scores for all categories
final['Final_Score'] = final.sum(axis=1)

# Display the final DataFrame
final
```

```
C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\2619076866.py:5:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_quiz['Weighted_Quiz'] = new_quiz['Total Quiz Score'] * weights['Total Quiz
```

```
Score']
C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\2619076866.py:6:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_homework['Weighted_HW'] = new_homework['Total Homework Score'] *
weights['Total Homework Score']
C:\Users\Mark Lopes\AppData\Local\Temp\ipykernel_16992\2619076866.py:7:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new_exam['Weighted_Exam'] = new_exam['Total Exam Score'] * weights['Total Exam
Score']
```

[ ]:

	Weighted_Quiz	Weighted_HW	Weighted_Exam	Final_Score
0	0.240	0.108	0.184	0.532
1	0.264	0.144	0.352	0.760
2	0.288	0.216	0.336	0.840