mod4_assigment-uc2

November 3, 2024

1 Module4: Numpy, Pandas, Matplotlib

1.1 Assignment: Use-Case II

[27]:
```
#1  Load the Data
import pandas as pd
from matplotlib import pyplot as plt

# Load the data
ds_data = pd.read_csv('DSScoreTerm1.csv')
maths_data = pd.read_csv('MathScoreTerm1.csv')
physics_data = pd.read_csv('PhysicsScoreTerm1.csv')

# Print basic information about the data (Optional)
print("\nData-Strucutre:\n-------------  ")
print(ds_data.info())
print(ds_data.head())

print("\nMaths:\n-------------  ")
print(maths_data.info())
print(maths_data.head())

print("\nPhysics:\n------------- ")
print(physics_data.info())
print(physics_data.head())
```

```
Data-Strucutre:
-------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 599 entries, 0 to 598
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Name        599 non-null    object
 1   Score       591 non-null    float64
 2   Age         599 non-null    int64
 3   Ethinicity  599 non-null    object
```

```
 4   Subject     599 non-null    object
 5   Sex         599 non-null    object
 6   ID          599 non-null    int64
dtypes: float64(1), int64(2), object(4)
memory usage: 32.9+ KB
None
             Name  Score  Age         Ethinicity        Subject Sex  ID
0  AI-KYUNG CHUNG   82.0   18     White American  Data Structue   M   1
1     ALAN HARVEY   79.0   19  European American  Data Structue   M   2
2    ALAN REYNAUD   39.0   19  European American  Data Structue   M   3
3  ALBERT CENDANA   76.0   18     White American  Data Structue   M   4
4  ALBERT HOLT JR   76.0   18     White American  Data Structue   F   5

Maths:
-------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 599 entries, 0 to 598
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Name        599 non-null    object
 1   Score       596 non-null    float64
 2   Age         599 non-null    int64
 3   Ethinicity  599 non-null    object
 4   Subject     599 non-null    object
 5   Sex         599 non-null    object
 6   ID          599 non-null    int64
dtypes: float64(1), int64(2), object(4)
memory usage: 32.9+ KB
None
             Name  Score  Age         Ethinicity Subject Sex  ID
0  AI-KYUNG CHUNG   88.0   18     White American   Maths   M   1
1     ALAN HARVEY   85.0   19  European American   Maths   M   2
2    ALAN REYNAUD   45.0   19  European American   Maths   M   3
3  ALBERT CENDANA   82.0   18     White American   Maths   M   4
4  ALBERT HOLT JR   82.0   18     White American   Maths   F   5

Physics:
-------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 599 entries, 0 to 598
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Name        599 non-null    object
 1   Score       593 non-null    float64
 2   Age         599 non-null    int64
 3   Ethinicity  599 non-null    object
```

```
 4   Subject     599 non-null    object
 5   Sex         599 non-null    object
 6   ID          599 non-null    int64
dtypes: float64(1), int64(2), object(4)
memory usage: 32.9+ KB
None
```

	Name	Score	Age	Ethinicity	Subject	Sex	ID
0	AI-KYUNG CHUNG	84.0	18	White American	Physics	M	1
1	ALAN HARVEY	81.0	19	European American	Physics	M	2
2	ALAN REYNAUD	41.0	19	European American	Physics	M	3
3	ALBERT CENDANA	78.0	18	White American	Physics	M	4
4	ALBERT HOLT JR	78.0	18	White American	Physics	F	5

[28]:
```
#2 Remove Confidential Columns: Drop the Name and Ethnicity columns to maintain␣
↪confidentiality.

# Remove 'Name' and 'Ethnicity' columns
ds_data = ds_data.drop(columns=['Name', 'Ethinicity'])
maths_data = maths_data.drop(columns=['Name', 'Ethinicity'])
physics_data = physics_data.drop(columns=['Name', 'Ethinicity'])
```

[29]:
```
#3 Fill Missing Score Data: Replace any missing Score values with 0.

# Fill missing 'Score' values with 0
ds_data['Score'] = ds_data['Score'].fillna(0)
maths_data['Score'] = maths_data['Score'].fillna(0)
physics_data['Score'] = physics_data['Score'].fillna(0)
```

[30]:
```
#4 Merge the Files: Combine the three datasets based on the common column ID.

# Rename 'Score' column to reflect each subject for clarity
ds_data = ds_data.rename(columns={'Score': 'DS_Score'})
maths_data = maths_data.rename(columns={'Score': 'Math_Score'})
physics_data = physics_data.rename(columns={'Score': 'Physics_Score'})

# Drop the 'Subject' column as it is redundant
ds_data = ds_data.drop(columns=['Subject'])

# Merge data on 'ID' (each dataset contains the same students)
merged_data = ds_data.merge(maths_data[['ID', 'Math_Score']], on='ID').
↪merge(physics_data[['ID', 'Physics_Score']], on='ID')
```

[31]:
```
#5 Convert Sex Column: Change Sex column values to 1 for males and 2 for␣
↪females.
# Convert 'Sex' column from 'M'/'F' to 1/2
merged_data['Sex'] = merged_data['Sex'].map({'M': 1, 'F': 2})
```

[32]:
```
#6 Save the Processed Data: Write the final cleaned and merged data to a new␣
 ↪CSV file, ScoreFinal.csv.
# Save the final processed data to 'ScoreFinal.csv'
merged_data.to_csv('ScoreFinal.csv', index=False)
# Print info and first few rows of the merged data for confirmation
print("Data has been processed and saved to 'ScoreFinal.csv'")
print(merged_data.info())
print(merged_data.head())
```

```
Data has been processed and saved to 'ScoreFinal.csv'
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 599 entries, 0 to 598
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   DS_Score       599 non-null    float64
 1   Age            599 non-null    int64
 2   Sex            599 non-null    int64
 3   ID             599 non-null    int64
 4   Math_Score     599 non-null    float64
 5   Physics_Score  599 non-null    float64
dtypes: float64(3), int64(3)
memory usage: 28.2 KB
None
   DS_Score  Age  Sex  ID  Math_Score  Physics_Score
0      82.0   18    1   1        88.0           84.0
1      79.0   19    1   2        85.0           81.0
2      39.0   19    1   3        45.0           41.0
3      76.0   18    1   4        82.0           78.0
4      76.0   18    2   5        82.0           78.0
```