

THỰC HÀNH LAB 03: LÀM SẠCH DỮ LIỆU CƠ BẢN

Vấn đề 1: Tiến hành tải dữ liệu vào chương trình ứng dụng Python và giải quyết vấn đề “Missing header in the csv file”

```
# Vấn đề 1: Xử lý thiếu header khi đọc file CSV
file_path = "patient_heart_rate.csv"
column_names = ["Id", "Name", "Age", "Weight", "m0006",
                "m0612", "m1218", "f0006", "f0612", "f1218"]
df = pd.read_csv(file_path, names=column_names)

# Hiển thị dữ liệu sau khi thêm header
print("Dữ liệu sau khi thêm header:")
print(df.head())
```

Dữ liệu sau khi thêm header:

	Id	Name	Age	Weight	m0006	m0612	m1218	f0006	f0612	f1218
0	1	Mickéy Mousé	56.0	70kgs	72	69	71	-	-	-
1	2	Donald Duck	34.0	154.89lbs	-	-	-	85	84	76
2	3	Mini Mouse	16.0	NaN	-	-	-	65	69	72
3	4	Scrooge McDuck	NaN	78kgs	78	79	72	-	-	-
4	5	Pink Panther	54.0	198.658lbs	-	-	-	69	NaN	75

Vấn đề 2: Xử lý vấn đề một cột lưu hỗn hợp nhiều dữ liệu, ở đây là cột “Name” chứa bao gồm “Firstname” và “Lastname”, giải pháp là ta sẽ tách ra làm 2 cột

```
# Vấn đề 2: Tách cột Name thành Firstname và Lastname
df[['Firstname', 'Lastname']] = df['Name'].str.split(expand=True, n=1)

# Hiển thị dữ liệu sau khi tách tên
print("Dữ liệu sau khi tách Firstname và Lastname:")
print(df.head())
```

Dữ liệu sau khi tách Firstname và Lastname:

	Id	Name	Age	Weight	m0006	m0612	m1218	f0006	f0612	f1218	Firstname	Lastname
0	1	Mickéy Mousé	56.0	70kgs	72	69	71	-	-	-	Mickéy	Mousé
1	2	Donald Duck	34.0	154.89lbs	-	-	-	85	84	76	Donald	Duck
2	3	Mini Mouse	16.0	NaN	-	-	-	65	69	72	Mini	Mouse
3	4	Scrooge McDuck	NaN	78kgs	78	79	72	-	-	-	Scrooge	McDuck
4	5	Pink Panther	54.0	198.658lbs	-	-	-	69	NaN	75	Pink	Panther

Vấn đề 3: Cột Weight có vấn đề về không thống nhất các đơn vị đo lường trong dữ liệu. Ta sẽ chuyển các đơn vị về thành đơn vị chuẩn “kg”

```
# Vấn đề 3: Xử lý không thống nhất đơn vị trong cột Weight
weight = df['Weight']

for i in range(len(weight)):
    x = str(weight[i])
    if "lbs" in x[-3:]:
        x = x[:-3] # Loại bỏ đơn vị "lbs"
        float_x = float(x) # Chuyển đổi thành số thực
        y = int(float_x / 2.2) # Chuyển đổi sang kg
        y = str(y) + "kgs" # Gán lại đơn vị "kgs"
        weight[i] = y

df['Weight'] = weight

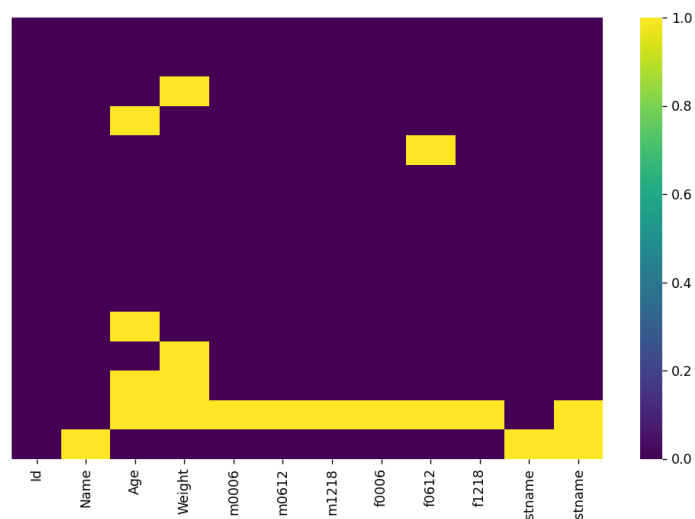
# Hiển thị dữ liệu sau khi chuẩn hóa trọng lượng
print("Dữ liệu sau khi chuẩn hóa đơn vị Weight:")
print(df.head())
```

Dữ liệu sau khi chuẩn hóa đơn vị Weight:

	Id	Name	Age	Weight	m0006	m0612	m1218	f0006	f0612	f1218	Firstname	Lastname
0	1	Mickéy Mousé	56.0	70kgs	72	69	71	-	-	-	Mickéy	Mousé
1	2	Donald Duck	34.0	70kgs	-	-	-	85	84	76	Donald	Duck
2	3	Mini Mouse	16.0	NaN	-	-	-	65	69	72	Mini	Mouse
3	4	Scrooge McDuck	NaN	78kgs	78	79	72	-	-	-	Scrooge	McDuck
4	5	Pink Panther	54.0	90kgs	-	-	-	69	NaN	75	Pink	Panther

Vấn đề 4: Vấn đề về xuất hiện dòng dữ liệu rỗng (không có giá trị: NaN). Giải pháp có thể đưa ra là xóa bỏ

```
# Vấn đề 4: Trực quan hóa dữ liệu bị thiếu bằng heatmap
plt.figure(figsize=(10, 6)) # Tùy chỉnh kích thước biểu đồ
sns.heatmap(df.isna(),
             yticklabels=False,
             cbar=True,
             cmap='viridis')
plt.show()
```



Vấn đề 5: Có nhiều dòng dữ liệu bị trùng lặp thông tin hoàn toàn[fullname, lastname, age, weight,...], giải pháp đưa ra là chỉ giữ lại một dòng dữ liệu, tuy nhiên giải pháp phải dựa trên nghiệp vụ của tập dữ liệu và quan sát của người xử lý.

```
# Vấn đề 5: Xóa các dòng dữ liệu trùng lặp dựa trên Firstname, Lastname, Age, Weight
df = df.drop_duplicates(subset=['Firstname', 'Lastname', 'Age', 'Weight'])
# Hiển thị dữ liệu sau khi loại bỏ trùng lặp
print("Dữ liệu sau khi loại bỏ các dòng trùng lặp:")
print(df.head(17))
```

Dữ liệu sau khi loại bỏ các dòng trùng lặp:

	Id	Age	Weight	m0006	m0612	m1218	f0006	f0612	f1218	Firstname	Lastname
0	1	56.0	70kgs	72	69	71	-	-	-	Mickéy	Mousé
1	2	34.0	70kgs	-	-	-	85	84	76	Donald	Duck
2	3	16.0	NaN	-	-	-	65	69	72	Mini	Mouse
3	4	NaN	78kgs	78	79	72	-	-	-	Scrooge	McDuck
4	5	54.0	90kgs	-	-	-	69	NaN	75	Pink	Panther
5	6	52.0	85kgs	-	-	-	68	75	72	Huey	McDuck
6	7	19.0	56kgs	-	-	-	71	78	75	Dewey	McDuck
7	8	32.0	78kgs	78	76	75	-	-	-	Scööpy	Doo
9	10	12.0	45kgs	-	-	-	92	95	87	Louie	McDuck
10	11	NaN	60kgs	78	75	72	-	-	-	Henry	Nam
11	12	34.0	NaN	65	67	55	-	-	-	Michel	Long
12	13	NaN	NaN	-	-	-	68	72	70	Tana	Ricky
13	14	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	UniLever	None
14	15	52.0	81kgs	-	-	-	68	75	72	NaN	NaN

Vấn đề 6: Xuất hiện dữ liệu bị ảnh hưởng bởi lỗi non-ASCII, không định dạng ASCII. Giải pháp: Tùy vào nghiệp vụ ta có thể: xóa dữ liệu tại đó, thay thế bằng dữ liệu khác hoặc thay bằng việc đánh dấu bằng một ký tự khác (ví dụ: 'warning')

```
# Vấn đề 6: Xử lý dữ liệu non-ASCII trong Firstname và Lastname
df.Firstname.replace({r'[^\\x00-\\x7F]+' : ''}, regex=True, inplace=True)
df.Lastname.replace({r'[^\\x00-\\x7F]+' : ''}, regex=True, inplace=True)

# Hiển thị dữ liệu sau khi xử lý non-ASCII
print("Dữ liệu sau khi loại bỏ ký tự non-ASCII:")
print(df.head())
```

Dữ liệu sau khi loại bỏ ký tự non-ASCII:

	Id	Age	Weight	m0006	m0612	m1218	f0006	f0612	f1218	Firstname	Lastname
0	1	56.0	70kgs	72	69	71	-	-	-	Micky	Mous
1	2	34.0	70kgs	-	-	-	85	84	76	Donald	Duck
2	3	16.0	NaN	-	-	-	65	69	72	Mini	Mouse
3	4	NaN	78kgs	78	79	72	-	-	-	Scrooge	McDuck
4	5	54.0	90kgs	-	-	-	69	NaN	75	Pink	Panther

Các dòng có Age và Weight bị thiếu (trừ ID 4 và 5): [10, 12, 13]

Vấn đề 7: “Missing values”, vấn đề này xảy ra tại các cột “Age”, “Weight” và “Heart Rate”. Thiếu dữ liệu (dữ liệu không đầy đủ) là vấn đề xảy ra nhiều trong các nguồn dữ liệu do nhiều nguyên nhân chủ quan lẫn khách quan. Có một vài giải pháp để xử lý vấn đề này, chủ yếu dựa trên kinh nghiệm và nghiệp vụ về tập dữ liệu đó. Một số giải pháp đưa đề xuất từ chuyên gia như sau:

- Deletion: Remove records with missing values
- Dummy substitution: Replace missing values with a dummy but valid value:
e.g.: 0 for numerical values.
- Mean substitution: Replace the missing values with the mean.
- Frequent substitution: Replace the missing values with the most frequent item.
- Improve the data collector: Your business folk will talk to the clients and inform them about why it is worth fixing the problem with the data collector.

Yêu cầu:

- Thống kê thông tin dữ liệu thiếu trên từng biến Age và Weight
- Yêu cầu xử lý dữ liệu thiếu như sau: Nếu dòng nào có Age hoặc Weight có dữ liệu thì phần Age hoặc Weight được tính như bên dưới, nếu thiếu cả 2 thông tin thì xóa dòng
- o Age: Giá trị thay thế là mean của các giá trị trong cột Age

```
# Vấn đề 7: Kiểm tra và xử lý dữ liệu thiếu
df['Age'] = pd.to_numeric(df['Age'], errors='coerce')
df['Weight'] = pd.to_numeric(df['Weight'], errors='coerce')
sns.heatmap(df[['Age']].isna(), yticklabels=False, cbar=True, cmap='viridis')
plt.show()

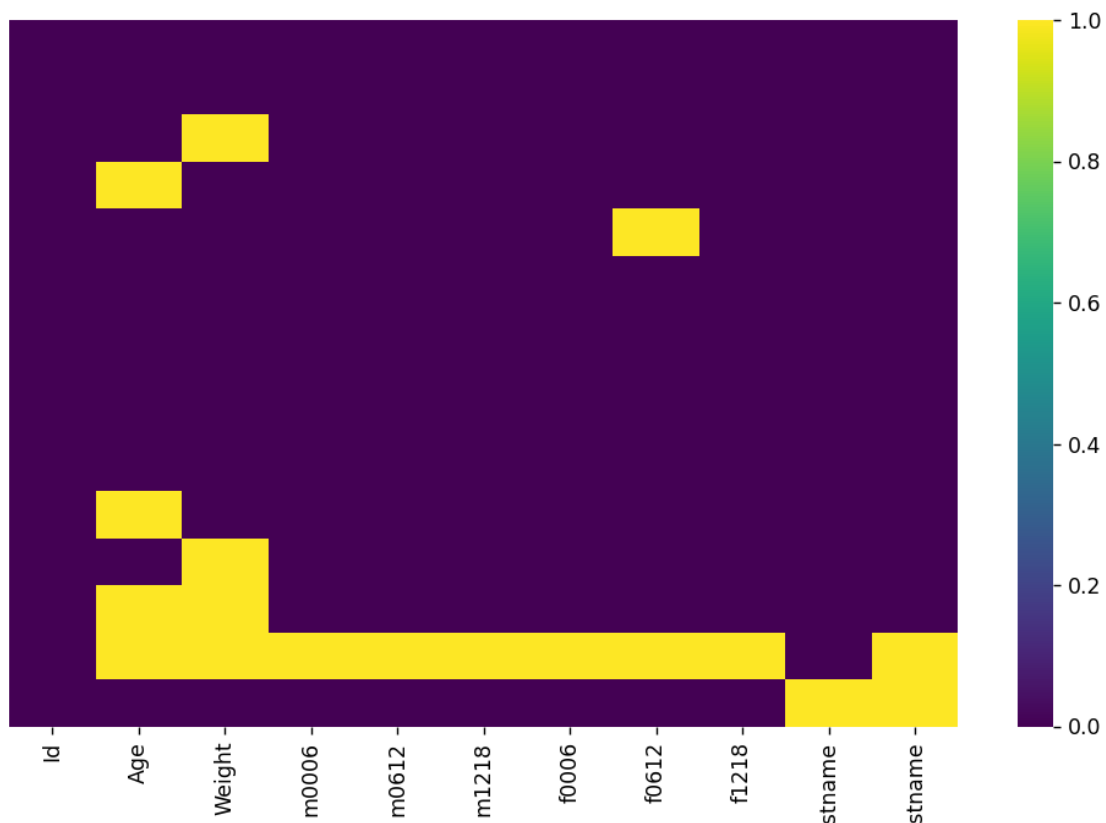
# Lấy index của các dòng thiếu cả Age và Weight, nhưng giữ lại ID 4 và 5
index_ageweightnull = df[(df['Age'].isna() & df['Weight'].isna()) & (~df['Id'].isin([4, 5]))].index
print("Các dòng có Age và Weight bị thiếu (trừ ID 4 và 5):", index_ageweightnull.tolist())

# Xóa các dòng thiếu cả Age và Weight ngoại trừ ID 4 và 5
df.drop(index=index_ageweightnull, inplace=True)

# Thay thế giá trị thiếu trong Age bằng giá trị trung bình
df['Age'] = df['Age'].fillna(df['Age'].mean())

# Thay thế giá trị thiếu trong Weight bằng giá trị trung bình
df['Weight'] = df['Weight'].fillna(df['Weight'].mean())

# Hiển thị dữ liệu sau khi xử lý missing values
print("Dữ liệu sau khi xử lý giá trị thiếu:")
print(df[['Age']].head())
```



```
Dữ liệu sau khi xử lý giá trị thiếu:
Age
0  56.0
1  34.0
2  16.0
3  36.1
4  54.0
```

Vấn đề 8: “một cột chứa quá nhiều thông tin cần được phân rã”, như trong bài toán này ta thấy header “m0006” chứa các nội dung bao gồm: m → male, 1218 ~ 12-18 (mm-dd).

```
# Vấn đề 8: Tách cột chứa giới tính, thời gian và nhịp tim
df = pd.melt(df, id_vars=['Id', 'Age', 'Weight', 'Firstname', 'Lastname'],
            value_name='PulseRate', var_name='sex_and_time').sort_values(['Id', 'Age', 'Weight', 'Firstname',
            'Lastname'])

# Tách Sex và Time từ sex_and_time
tmp_df = df["sex_and_time"].str.extract(r'([mf])(\d{2})(\d{2})', expand=True)
tmp_df.columns = ["Sex", "hours_lower", "hours_upper"]

# Tạo cột Time từ hours_lower và hours_upper
tmp_df["Time"] = tmp_df["hours_lower"] + "-" + tmp_df["hours_upper"]

# Gộp lại với dataframe chính
df = pd.concat([df, tmp_df], axis=1)

# Xóa các cột không cần thiết
df = df.drop(["sex_and_time", "hours_lower", "hours_upper"], axis=1)
df = df.dropna()

print("Dữ liệu sau khi tách cột Sex, Time và PulseRate:")
print(df.head())
df.to_csv("outputcleanup.csv", index=False)
```

Vấn đề 11: Hãy khảo sát tỉ lệ dữ liệu thiếu trên biến huyết áp. Dữ liệu bị thiếu thì hãy xử lý bằng phương pháp sau

- Thay thế bằng giá trị trung bình liền trước và liền sau của người đó. Nếu không được thì dùng 2)
- Thay thế bằng giá trị trung bình 2 giá liền trước của người đó. Nếu không được thì dùng 3)
- Thay thế bằng giá trị trung bình 2 giá liền sau của người đó. Nếu không được thì dùng 4)
- Trung bình của các giá trị huyết áp của người đó. Nếu không được thì dùng 5).
- Trung bình của các giá trị huyết áp của nhóm giới tính. Nếu không được thì dùng 6)
- Trung bình của các giá trị dữ liệu. Nếu không được thì thay bằng mức ổn định trong y học.

```
# Vấn đề 11: Xử lý dữ liệu thiếu trên biến huyết áp
def avgPulserate(df, firstname):
    valid_pulses = df.loc[(df['Firstname'] == firstname) & (df['PulseRate'] != 0), 'PulseRate']
    return valid_pulses.mean() if not valid_pulses.empty else 0

def isValidIndex1(index, nrows):
    return index - 1 >= 0 and index + 1 < nrows

def isValidIndex2(index, nrows):
    return index + 2 < nrows

def isPulseRateNotNull1(index, df):
    return df.at[index - 1, 'PulseRate'] != 0 and df.at[index + 1, 'PulseRate'] != 0

def isPulseRateNotNull2(index, df):
    return df.at[index + 1, 'PulseRate'] != 0 and df.at[index + 2, 'PulseRate'] != 0

def isInGroup1(df, index, firstname):
    return df.at[index - 1, 'Firstname'] == firstname and df.at[index + 1, 'Firstname'] == firstname

def isInGroup2(df, index, firstname):
    return df.at[index + 1, 'Firstname'] == firstname and df.at[index + 2, 'Firstname'] == firstname

def updatePulseRateNa(index, df):
    nrows = len(df['PulseRate'])
    firstname = df.at[index, 'Firstname']
    new_value = 0
    try:
        if isValidIndex1(index, nrows) and isPulseRateNotNull1(index, df) and isInGroup1(df, index, firstname):
            new_value = (df.at[index - 1, 'PulseRate'] + df.at[index + 1, 'PulseRate']) / 2
        elif isValidIndex2(index, nrows) and isPulseRateNotNull2(index, df) and isInGroup2(df, index, firstname):
            new_value = (df.at[index + 1, 'PulseRate'] + df.at[index + 2, 'PulseRate']) / 2
        else:
            new_value = avgPulserate(df, firstname)
    except:
        new_value = avgPulserate(df, firstname)
    return new_value

pulse_rate_update = [updatePulseRateNa(i, df) if v == 0 else v for i, v in enumerate(df['PulseRate'])]
df['PulseRate'] = pulse_rate_update
```


Vấn đề 12: Hãy rút gọn dữ liệu phù hợp và reindex lại dữ liệu. Sau đó, lưu trữ dữ liệu đã xử lý thành công với tên file patient_heart_rate_clean.csv

Dữ liệu sau khi tách cột Sex, Time và PulseRate:

Empty DataFrame

Columns: [Id, Age, Weight, Firstname, Lastname, PulseRate, Sex, Time]

Index: []

 outputcleanup.csv