Ôn tập giữa kỳ

```
# import library...
# ...
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

ĐỀ BÀI

Hệ thống giám sát rủi ro dựa trên hành vi (The Behavior Risk Factor Surveilance System - BRFSS) là một cuộc khảo sát qua điện thoại hàng năm với 350.000 người ở Hoa Kỳ. Như tên gọi của nó, BRFSS được thiết kế để xác định các yếu tố nguy cơ ở người trưởng thành và báo cáo các xu hướng sức khỏe mớ. Ví dụ, người trả lời được hỏi về chế độ ăn uống và hoạt động thể chất hàng tuần, tình trạng HIV/AIDS, khả năng sử dụng thuốc lá và thậm chí cả mức độ chi trả dịch vụ chăm sóc sức khỏe của họ.

Bộ dữ liệu **brfss_2000** chứa thông tin khảo sát năm 2000, với hơn 200 thông tin. Trong bộ dữ liệu này, ta chỉ khảo sát một số thông tin sau:

- genhlth: người khảo sát tự đánh giá sức khỏe (excellent, very good, good, fair or poor)
- exerany: cho biết có hoạt động thể chất nào trong tháng gần nhất hay không, có (1), không (0)
- hlthplan: có bảo hiểm (1) hay không (0)
- smoke100: tổng số điếu thuốc đã hút ít nhất
- height: chiều cao (inches)
- weight: cân nặng (pounds)
- wtdesire: cân nặng mong muốn(pounds)
- age: tuổi
- gender: giới tính: nam(m), nữ(f)

Câu 1:

Đọc hai bộ dữ liệu trên và cho biết mỗi bộ dữ liệu có kích thước bao nhiêu dòng, bao nhiêu cột?

```
df = pd.read_csv('brfss_2000 .csv')
df
```

	exerany	hlthplan	smoke100	height	weight	wtdesire	age	gender	genhlth
0	0	1	0	70	175	175	77	m	good
1	0	1	1	64	125	115	33	f	good
2	1	1	1	60	105	105	49	f	good
3	1	1	0	66	132	124	42	f	good
4	0	1	0	61	150	130	55	f	very good
...	...	...	...	...	...	...	...	...	...
19995	1	1	0	66	215	140	23	f	good
19996	0	1	0	73	200	185	35	m	excellent
19997	0	1	0	65	216	150	57	f	poor
19998	1	1	0	67	165	165	81	f	good
19999	1	1	1	69	170	165	83	m	good

20000 rows × 9 columns

Câu 2

a, Tính tỷ lệ nam nữ

+ Mã + Văn bản

```
# Tính số lượng nam và nữ
```

```
gender_counts = df['gender'].value_counts()

# Tính tỷ lệ nam và nữ
gender_ratio = gender_counts / gender_counts.sum()
print(gender_ratio)
```

```
gender
f    0.52155
m    0.47845
Name: count, dtype: float64
```

b, Trong số những người tập thể thao, tỷ lệ những người tự đánh giá có sức khỏe kém là bao nhiêu?

```
# Lọc những người tập thể thao
exercise_group = df[df['exerany'] == 1]

# Tính tỷ lệ người có sức khỏe kém trong nhóm tập thể thao
poor_health_ratio = (exercise_group['genhlth'] == 'poor').mean()
print(poor_health_ratio)
```

```
0.01964597022931474
```

Câu 3

a, Đổi đơn vị chiều cao từ inches sang centimet, đơn vị cân nặng từ pound sang kg.

```
# Đổi đơn vị chiều cao từ inches sang centimet
df['height_cm'] = df['height'] * 2.54

# Đổi đơn vị cân nặng từ pounds sang kilogram
df['weight_kg'] = df['weight'] * 0.453592

# Hiển thị các cột mới
print(df[['height', 'height_cm', 'weight', 'weight_kg']].head())
```

```
   height  height_cm  weight  weight_kg
0      70     177.80     175  79.378600
1      64     162.56     125  56.699000
2      60     152.40     105  47.627160
3      66     167.64     132  59.874144
4      61     154.94     150  68.038800
```

b, Tính tỷ lệ những người muốn giảm cân.

```
# Tính tỷ lệ những người muốn giảm cân
weight_loss_ratio = (df['wtdesire'] < df['weight']).mean()
print(weight_loss_ratio)
```

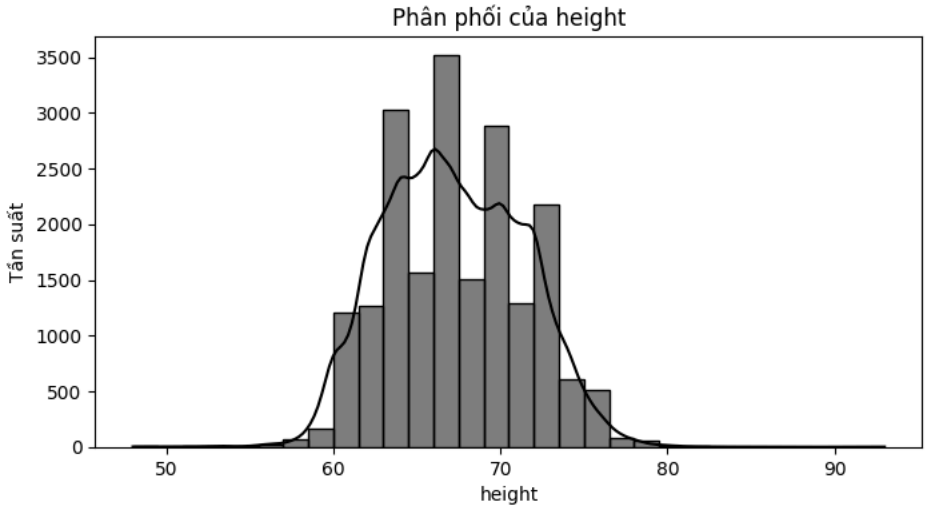
```
0.6382
```

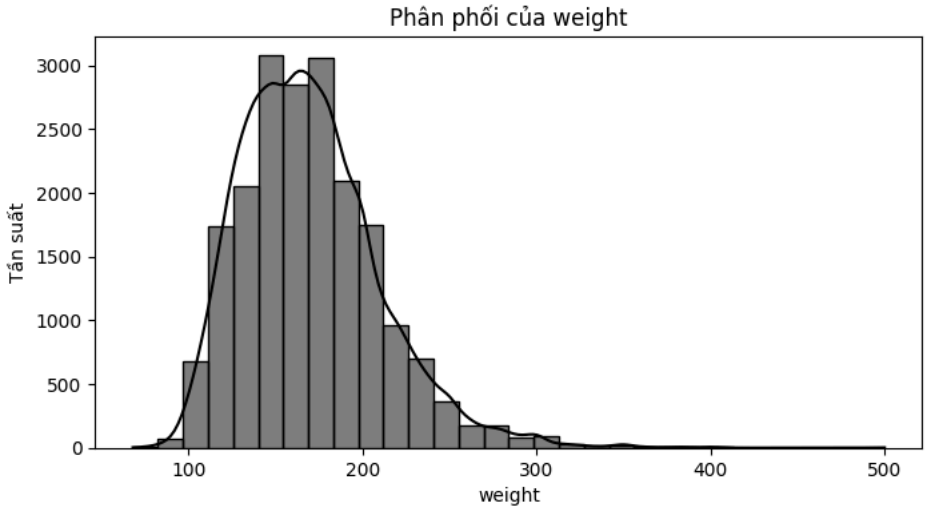
Câu 4:

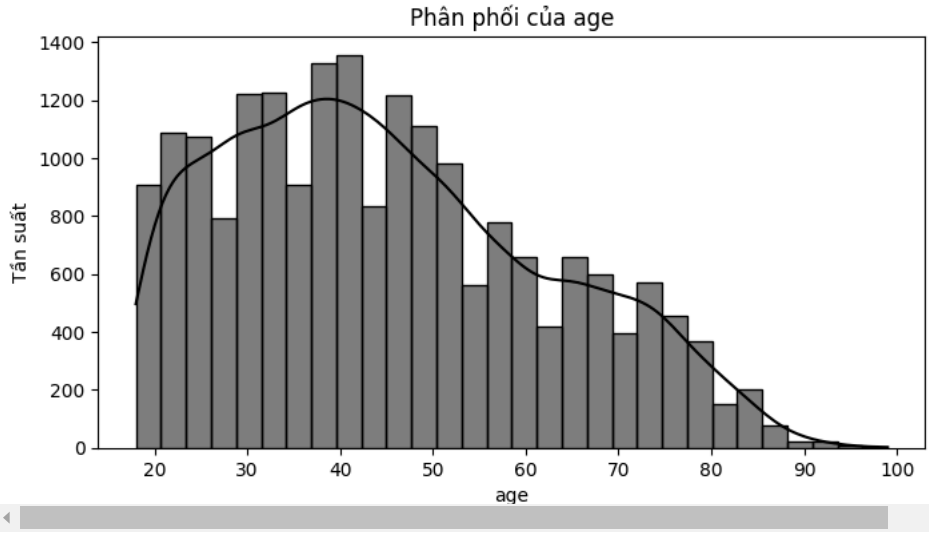
a, Theo bạn trong các thuộc tính trên, thuộc tính nào có phân phối chuẩn. Vẽ hình minh họa

```
# Vẽ histogram cho các thuộc tính
columns_to_check = ['height', 'weight', 'age']
for col in columns_to_check:
    plt.figure(figsize=(8, 4))
    sns.histplot(df[col], kde=True, bins=30, color='black')
    plt.title(f'Phân phối của {col}')
    plt.xlabel(col)
    plt.ylabel('Tần suất')
    plt.show()
```

[+]

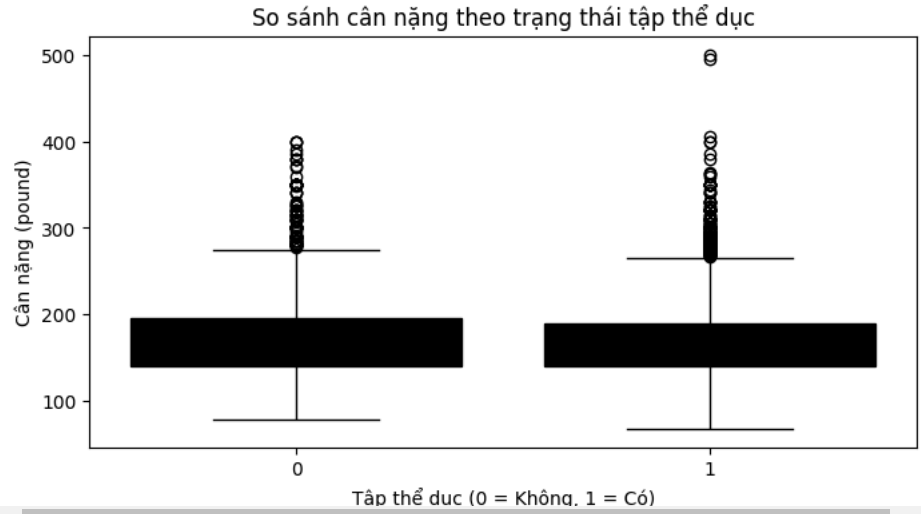






b, Vẽ đồ thị boxplot so sánh cân nặng của những người có tập thể dục

```
# Vẽ boxplot so sánh cân nặng giữa những người có tập thể dục và không tập
plt.figure(figsize=(8, 4))
sns.boxplot(x='exerany', y='weight', data=df, color='black')
plt.title('So sánh cân nặng theo trạng thái tập thể dục')
plt.xlabel('Tập thể dục (0 = Không, 1 = Có)')
plt.ylabel('Cân nặng (pound)')
plt.show()
```



Câu 5

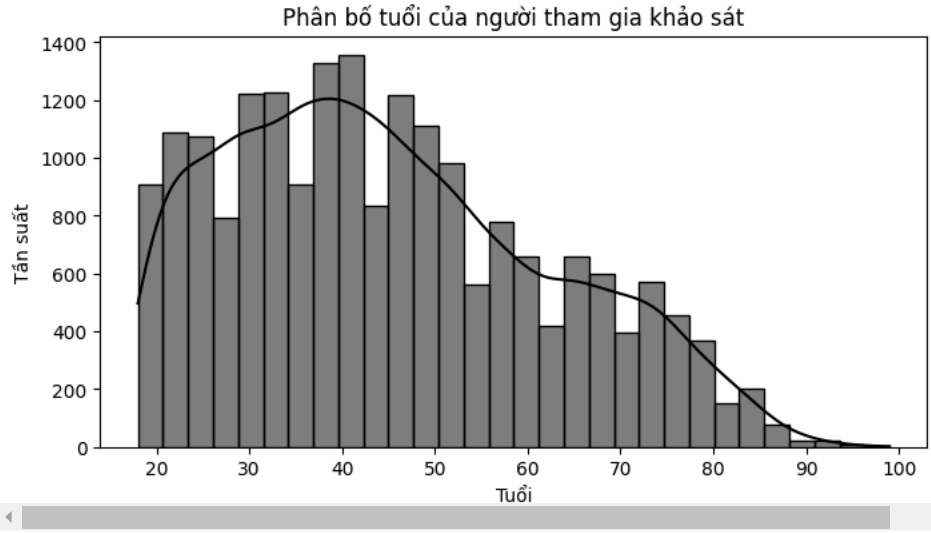
Phân bố tuổi tác trong mẫu: Hãy mô tả phân bố tuổi của người tham gia khảo sát. Tuổi trung bình, độ lệch chuẩn, và các phân vị 25%, 50%, 75% là bao nhiêu?

```
# Tính các thông số mô tả phân bố tuổi
age_mean = df['age'].mean()
age_std = df['age'].std()
age_quantiles = df['age'].quantile([0.25, 0.5, 0.75])

print(f"Tuổi trung bình: {age_mean:.2f}")
print(f"Độ lệch chuẩn: {age_std:.2f}")
print("Các phân vị:")
print(age_quantiles)

# Vẽ biểu đồ phân bố tuổi
plt.figure(figsize=(8, 4))
sns.histplot(df['age'], bins=30, kde=True, color='black')
plt.title('Phân bố tuổi của người tham gia khảo sát')
plt.xlabel('Tuổi')
plt.ylabel('Tần suất')
plt.show()
```

```
Tuổi trung bình: 45.07
Độ lệch chuẩn: 17.19
Các phân vị:
0.25    31.0
0.50    43.0
0.75    57.0
Name: age, dtype: float64
```



Câu 6

Tỉ lệ người hút thuốc: Tính tỉ lệ phần trăm người tham gia khảo sát hiện đang hút thuốc lá. Liệu có sự khác biệt đáng kể về tỉ lệ này giữa các nhóm tuổi khác nhau không?

```
# Tính tỷ lệ phần trăm người hút thuốc
smoker_percentage = (df['smoke100'] == 1).mean() * 100
print(f"Tỷ lệ phần trăm người hút thuốc lá: {smoker_percentage:.2f}%")
```

```
Tỷ lệ phần trăm người hút thuốc lá: 47.21%
```

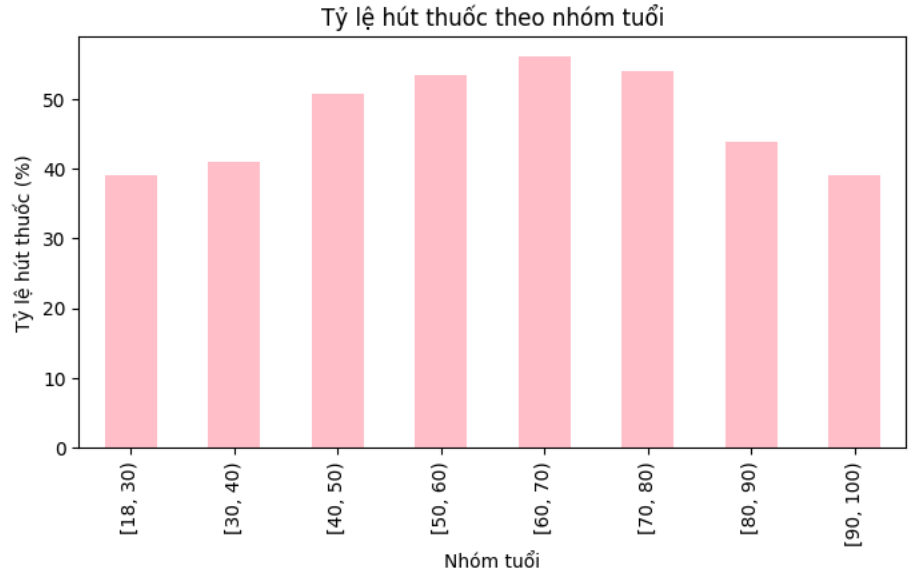
```
# Phân nhóm tuổi
bins = [18, 30, 40, 50, 60, 70, 80, 90, 100]
age_groups = pd.cut(df['age'], bins=bins, right=False)

# Tính tỷ lệ hút thuốc trong từng nhóm tuổi
smoking_by_age = df.groupby(age_groups)['smoke100'].mean() * 100

print(smoking_by_age)

plt.figure(figsize=(8,4))
smoking_by_age.plot(kind='bar', color='pink')
plt.title('Tỷ lệ hút thuốc theo nhóm tuổi')
plt.xlabel('Nhóm tuổi')
plt.ylabel('Tỷ lệ hút thuốc (%)')
plt.show()
```

```
age
[18, 30)     39.013558
[30, 40)     41.073102
[40, 50)     50.741912
[50, 60)     53.477380
[60, 70)     56.178707
[70, 80)     54.119062
[80, 90)     43.804538
[90, 100)    39.024390
Name: smoke100, dtype: float64
C:\Users\Admin\AppData\Local\Temp\ipykernel_28036\1787127265.py:6: FutureWarning: The default of observed=False is deprecated and will b
  smoking_by_age = df.groupby(age_groups)['smoke100'].mean() * 100
```



Câu 7

BMI trung bình theo giới tính: Tính chỉ số BMI trung bình cho nam và nữ trong mẫu.

```
# Tính chiều cao bằng mét
df['height_m'] = df['height_cm'] / 100  # Chuyển chiều cao từ cm sang m

# Tính chỉ số BMI
df['BMI'] = df['weight'] / (df['height_m'] ** 2)
```

```
# Tính BMI trung bình theo giới tính
bmi_mean_by_gender = df.groupby('gender')['BMI'].mean().reset_index()

print(bmi_mean_by_gender)
```

```
  gender        BMI
0      f  56.755133
1      m  59.362406
```

câu 8

Phân tích tỉ lệ bệnh béo phì: Sử dụng các tiêu chuẩn của CDC về chỉ số BMI để phân loại người tham gia vào nhóm béo phì. Tính tỉ lệ béo phì theo giới tính và độ tuổi.

```
# Tính chiều cao bằng mét
df['height_m'] = df['height_cm'] / 100  # Chuyển chiều cao từ cm sang m

# Tính chỉ số BMI
df['BMI'] = df['weight'] / (df['height_m'] ** 2)

# Phân loại người tham gia vào nhóm béo phì
df['obesity'] = df['BMI'] >= 30

# Tính tỷ lệ béo phì theo giới tính và độ tuổi
obesity_rate = df.groupby(['gender', 'age'])['obesity'].mean().reset_index()
obesity_rate['obesity_rate'] = obesity_rate['obesity'] * 100  # Chuyển sang tỷ lệ phần trăm

print(obesity_rate[['gender', 'age', 'obesity_rate']])
```

```
    gender  age  obesity_rate
0        f   18         100.0
1        f   19         100.0
2        f   20         100.0
3        f   21         100.0
4        f   22         100.0
..     ...  ...           ...
153      m   90         100.0
154      m   91         100.0
155      m   92         100.0
156      m   93         100.0
157      m   94         100.0

[158 rows x 3 columns]
```
