LAB 4: BÀI THỰC HÀNH CHUẨN BỊ DỮ LIỆU

PHẦN 1: DATA CLEANSING & FEATURE ENGINEERING

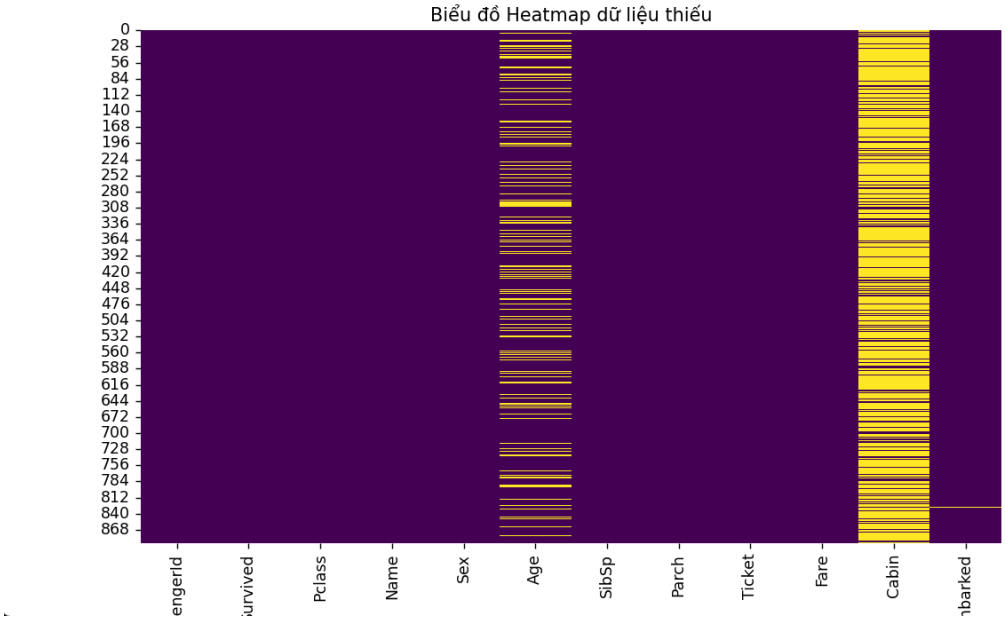
1. Viết hàm load_data() để tải dữ liệu lên ứng dụng. Sau đó, hiển thị ra màn hình 10 dòng đầu tiên.

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Bài 1
def load_data():
    df = pd.read_csv('titanic_disaster.csv')
    print(df.head(10))
    return df

df = load_data()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

2. Thống kê dữ liệu thiếu trên các biến số và trực quan hóa dữ liệu thiếu bằng biểu đồ (Heat map). Hãy cho nhận xét về tình trạng thiếu dữ liệu Age, Cabin và Embarked

```python
# Bài 2
# Thống kê và heatmap dữ liệu thiếu
print(df.isnull().sum())
plt.figure(figsize=(10, 6))
sns.heatmap(df.isnull(), cmap='viridis', cbar=False)
plt.title('Biểu đồ Heatmap dữ liệu thiếu')
plt.show()
```




```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

3. Xử lý tên cột tên Name, tách ra làm 2 cột: firstName và secondName. Lưu ý: Sau khi tách cột xong thì xóa luôn cột Name

```python
# Bài 3
# Tách tên thành lastName, firstName
df[['lastName', 'firstName']] = df['Name'].str.extract(r'([A-Za-z]+),\s(.+)')
df.drop('Name', axis=1, inplace=True)
```

```
Bài 3
    lastName                                          firstName
0     Braund                                  Mr. Owen Harris
1    Cumings  Mrs. John Bradley (Florence Briggs Thayer)
2  Heikkinen                                      Miss. Laina
3   Futrelle         Mrs. Jacques Heath (Lily May Peel)
4      Allen                               Mr. William Henry
5      Moran                                       Mr. James
6   McCarthy                                  Mr. Timothy J
7    Palsson                         Master. Gosta Leonard
8    Johnson    Mrs. Oscar W (Elisabeth Vilhelmina Berg)
9     Nasser                   Mrs. Nicholas (Adele Achem)
```

4. Xứ lý rút gọn kích thước dữ liệu trên cột Sex như sau: thay thế male → M và female → F

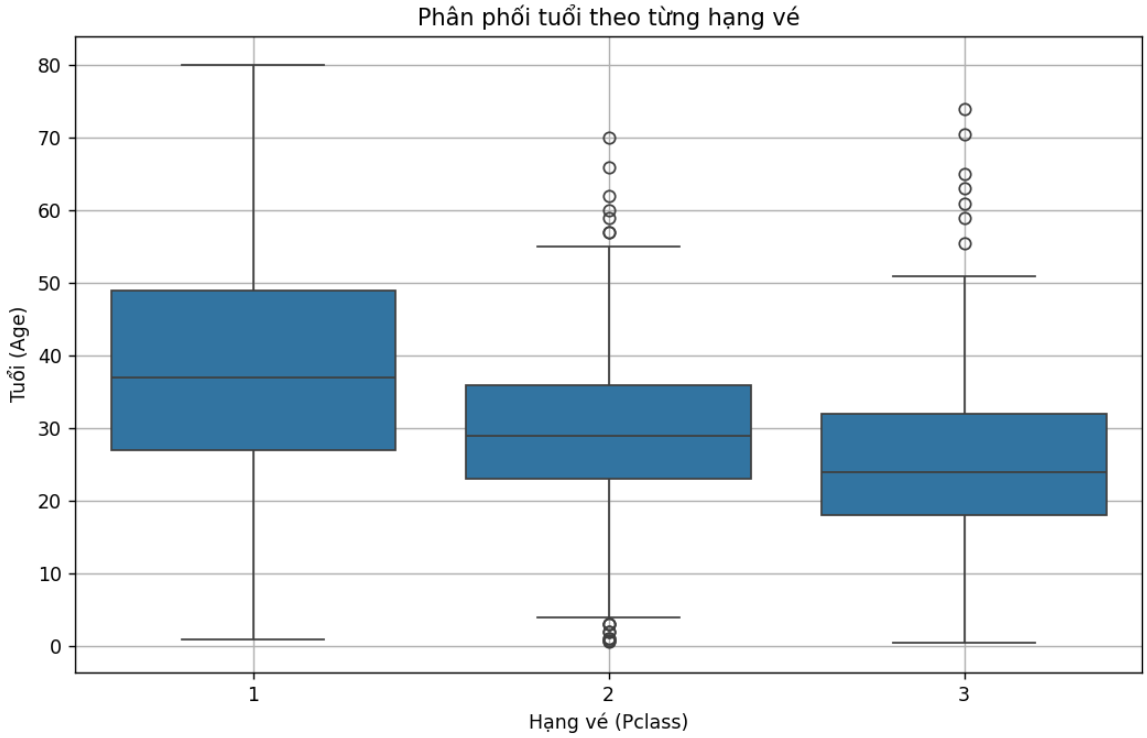
```python
# Bài 4
# Rút gọn dữ liệu cột Sex
df['Sex'] = df['Sex'].replace({'male': 'M', 'female': 'F'})

# Kiểm tra kết quả
print(df[['lastName', 'firstName', 'Sex']].head(10))
```

```
Bài 4
    lastName                                          firstName Sex
0     Braund                                Mr. Owen Harris   M
1    Cumings  Mrs. John Bradley (Florence Briggs Thayer)   F
2  Heikkinen                                    Miss. Laina   F
3   Futrelle         Mrs. Jacques Heath (Lily May Peel)   F
4      Allen                           Mr. William Henry   M
5      Moran                                   Mr. James   M
6   McCarthy                                Mr. Timothy J   M
7    Palsson                      Master. Gosta Leonard   M
8    Johnson    Mrs. Oscar W (Elisabeth Vilhelmina Berg)   F
9     Nasser                 Mrs. Nicholas (Adele Achem)   F
PassengerId      0
Survived         0
Pclass           0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
lastName         0
firstName        0
dtype: int64
```

5. Xử lý dữ liệu thiếu trên biến Age bằng cách thay thế bằng giá trị trung bình tuổi: Hãy đưa ra quyết định dùng giá trị trung bình tuổi toàn bộ hành khách hay theo từng nhóm hạng vé (hạng hành khách: Pclass). Ta tiến hành làm các bước sau:

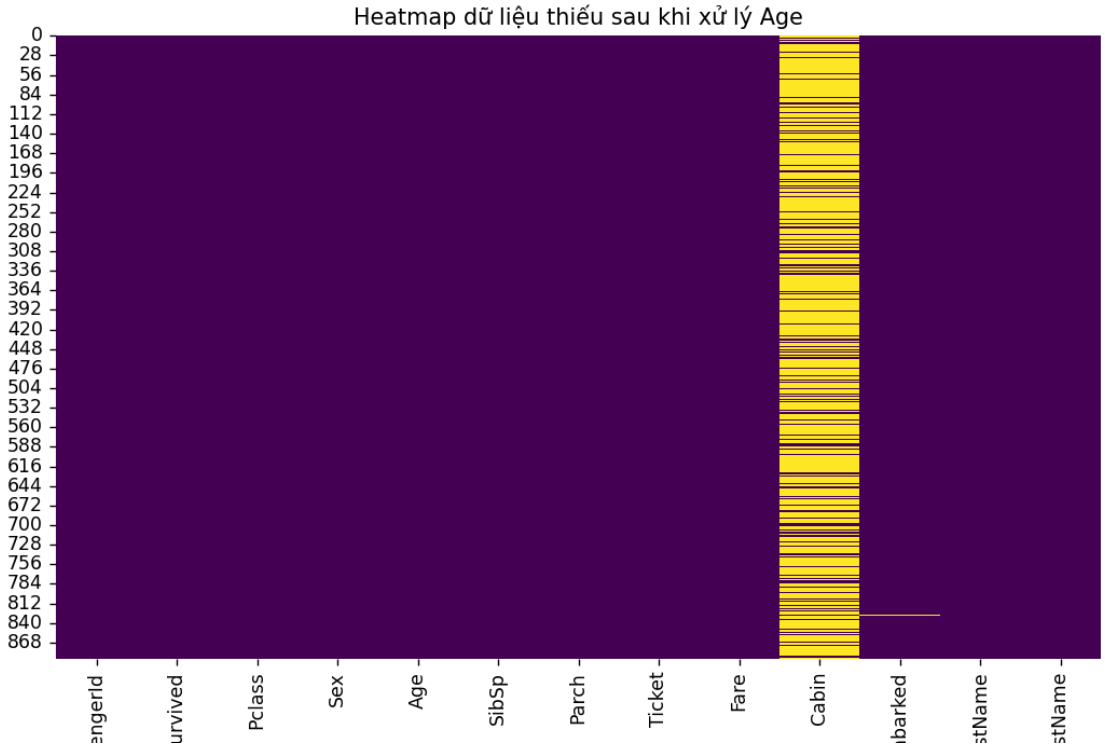
a. Sử dụng Seaborn để vẽ biểu đồ (Box plot) trực quan dữ liệu để xác định phân phối tuổi trên từng hạng hành khách. Nhận xét về tuổi trung bình giữa các nhóm hành khách. Từ đó đưa ra quyết định cách thay thế giá trị tuổi bị thiếu.

```python
# Bài 5
# a
plt.figure(figsize=(10, 6))
sns.boxplot(x='Pclass', y='Age', data=df)
plt.title('Phân phối tuổi theo từng hạng vé')
plt.xlabel('Hạng vé (Pclass)')
plt.ylabel('Tuổi (Age)')
plt.grid(True)
plt.show()

# Nhận xét
# Hạng 1: Tuổi trung bình cao hơn hẳn (khoảng trên 35).
# Hạng 2: Tuổi trung bình khoảng 29.
# Hạng 3: Tuổi trung bình thấp nhất, chỉ khoảng 22.
# → Quyết định: Điền dữ liệu thiếu theo trung vị (median) của từng Pclass thay vì toàn bộ dataset.
```



b. Tiến hành thay thế giá trị Age bị thiếu. Sau đó, hiển thị kết quả dạng bảng và trực quan dữ liệu đã xử lý thiếu cho cột’Age’ bằng biểu đồ Heat map.

```python
#b: Điền tuổi theo median của từng hạng vé
def impute_age(row):
    if pd.isnull(row['Age']):
        if row['Pclass'] == 1:
            return df[df['Pclass'] == 1]['Age'].median()
        elif row['Pclass'] == 2:
            return df[df['Pclass'] == 2]['Age'].median()
        else:
            return df[df['Pclass'] == 3]['Age'].median()
    else:
        return row['Age']

df['Age'] = df.apply(impute_age, axis=1)

print(df.isnull().sum())

plt.figure(figsize=(10, 6))
sns.heatmap(df.isnull(), cmap='viridis', cbar=False)
plt.title('Heatmap dữ liệu thiếu sau khi xử lý Age')
plt.show()
```



6. Xây dựng biến số Agegroup có thang đo thứ tự được ánh xạ theo thang đo khoảng dựa trên độ tuổi của hành khách như sau: (age =< 12] → Kid; (12, 18]: Teen, (18, 60]: Adult và (age > 60): Older

```python
# Bài 6
print("Bài 6")
def age_group(age):
    if age <= 12:
        return 'Kid'
    elif age <= 18:
        return 'Teen'
    elif age <= 60:
        return 'Adult'
    else:
        return 'Older'

df['AgeGroup'] = df['Age'].apply(age_group)

print(df[['Age', 'AgeGroup']].head(10))
```

```
Bài 6
    Age AgeGroup
0  22.0    Adult
1  38.0    Adult
2  26.0    Adult
3  35.0    Adult
4  35.0    Adult
5  24.0    Adult
6  54.0    Adult
7   2.0      Kid
8  27.0    Adult
9  14.0     Teen
```

7. Tiến hành thêm đặc trưng về danh xưng (namePrefix) trong xã hội bằng cách tách Mr, Mrs, Miss, Master ra khỏi “secondName”

```python
# Bài 7
print("Bài 7")
df['NamePrefix'] = df['firstName'].str.extract(r'([A-Za-z]+)\.')

print(df[['firstName', 'NamePrefix']].head(10))
```

```
Bài 7
                                        firstName NamePrefix
0                                Mr. Owen Harris         Mr
1  Mrs. John Bradley (Florence Briggs Thayer)        Mrs
2                                    Miss. Laina       Miss
3            Mrs. Jacques Heath (Lily May Peel)        Mrs
4                              Mr. William Henry         Mr
5                                    Mr. James         Mr
6                                  Mr. Timothy J         Mr
7                          Master. Gosta Leonard     Master
8       Mrs. Oscar W (Elisabeth Vilhelmina Berg)        Mrs
9                    Mrs. Nicholas (Adele Achem)        Mrs
```

8. Khai thác thêm thông tin số lượng thành viên đi theo nhóm thân quen (familySize) đối với mỗi hành khách trên chuyến hải trình; family size = 1+ SibSp + Parch

```python
# Bài 8
print("Bài 8")
df['FamilySize'] = df['SibSp'] + df['Parch'] + 1

print(df[['SibSp', 'Parch', 'FamilySize']].head(10))
```

```
Bài 8
   SibSp  Parch  FamilySize
0      1      0           2
1      1      0           2
2      0      0           1
3      1      0           2
4      0      0           1
5      0      0           1
6      0      0           1
7      3      1           5
8      0      2           3
9      1      0           2
```

9. Tạo thêm đặc trưng ‘Alone’ để xác định hành khách đi theo nhóm hay cá nhân bằng cách dựa trên familySize như sau: Nếu familySize = 0 thì giá trị Alone = 1 và ngược lại là 0.

```python
# Bài 9
print("Bài 9")
df['Alone'] = df['FamilySize'].apply(lambda x: 1 if x == 1 else 0)

print(df[['FamilySize', 'Alone']].head(10))
```

```
Bài 9
   FamilySize  Alone
0           2      0
1           2      0
2           1      1
3           2      0
4           1      1
5           1      1
6           1      1
7           5      0
8           3      0
9           2      0
```

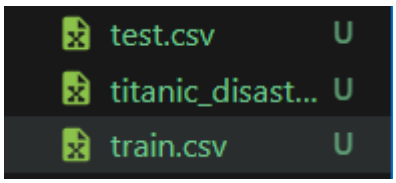
10. Tiến hành tách loại cabin (typeCabin) mà hành khách ở để lọc và phân tích đặc tính cabin. Loại cabin được kí hiệu bởi chữ cái đầu tiên. Lưu ý: Đối với dữ liệu cabin bị thiếu thì thay thế bằng “Unknown”

```python
# Bài 10
print("Bài 10")
def extract_cabin_type(cabin):
    if pd.isnull(cabin):
        return 'Unknown'
    else:
        return cabin[0]

df['typeCabin'] = df['Cabin'].apply(extract_cabin_type)

print(df[['Cabin', 'typeCabin']].head(10))
```

```
Bài 10
   Cabin typeCabin
0    NaN    Unknown
1    C85          C
2    NaN    Unknown
3   C123          C
4    NaN    Unknown
5    NaN    Unknown
6    E46          E
7    NaN    Unknown
8    NaN    Unknown
9    NaN    Unknown
```

11. Loại bỏ dữ liệu thừa đối với các hành khách xuất hiện trong cả 2 tập dữ liệu huấn luyện (train.csv) và đánh giá (test.csv). Ưu tiên giữ lại dữ liệu trong tập huấn luyện.

```python
# Bài 11
print("Bài 11")
from sklearn.model_selection import train_test_split

train, test = train_test_split(df, test_size=0.2, random_state=42)

train = train[~train['PassengerId'].isin(test['PassengerId'])]

train.to_csv('train.csv', index=False)
test.to_csv('test.csv', index=False)
```



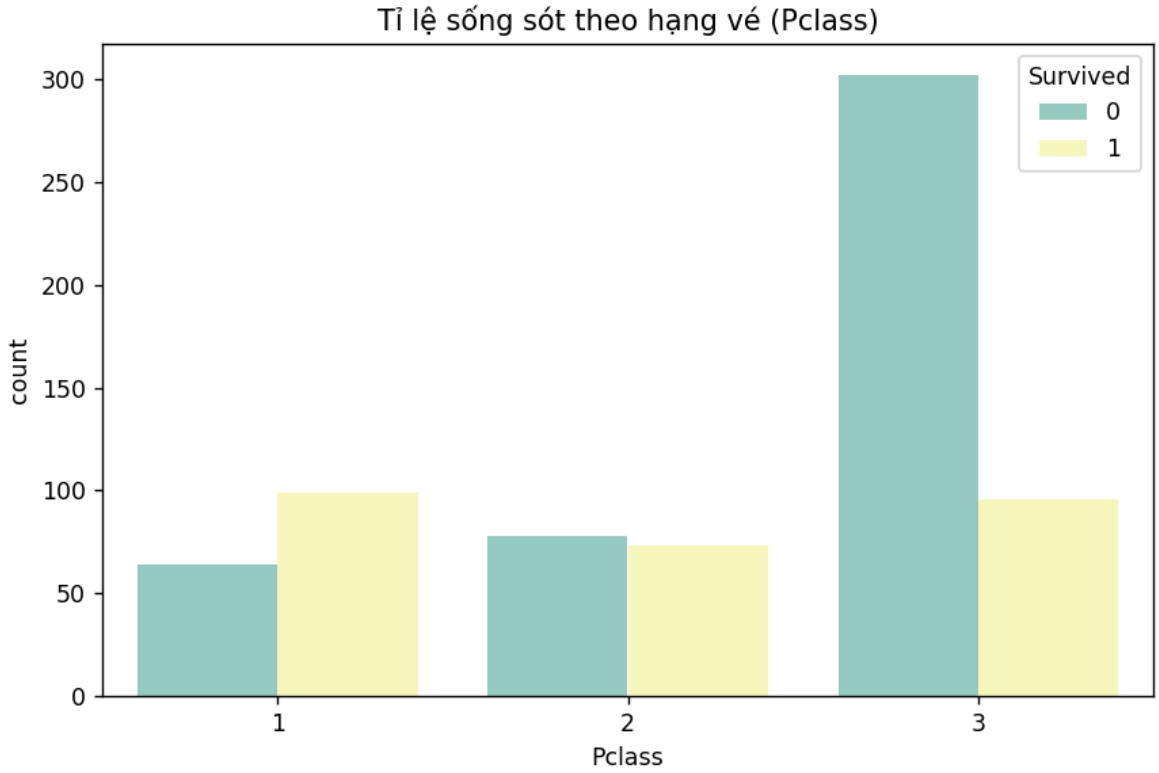
PHẦN 2: KHAI THÁC THÔNG TIN HỮU ÍCH – EDA

12. Trực quan thông tin tương quan tỉ lệ sống sót và thiệt mạng trên từng nhóm giới tính.

```
# Bài 12
print("Bài 12")

plt.figure(figsize=(8, 5))
sns.countplot(x='Sex', hue='Survived', data=train, palette='Set2')
plt.title('Tỉ lệ sống sót và thiệt mạng theo giới tính')
plt.show()

# Nhận xét:
# - Phụ nữ (female) có tỉ lệ sống sót cao hơn nam giới rất nhiều.
# - Điều này phù hợp với quy tắc "Phụ nữ và trẻ em lên thuyền cứu sinh trước".
```

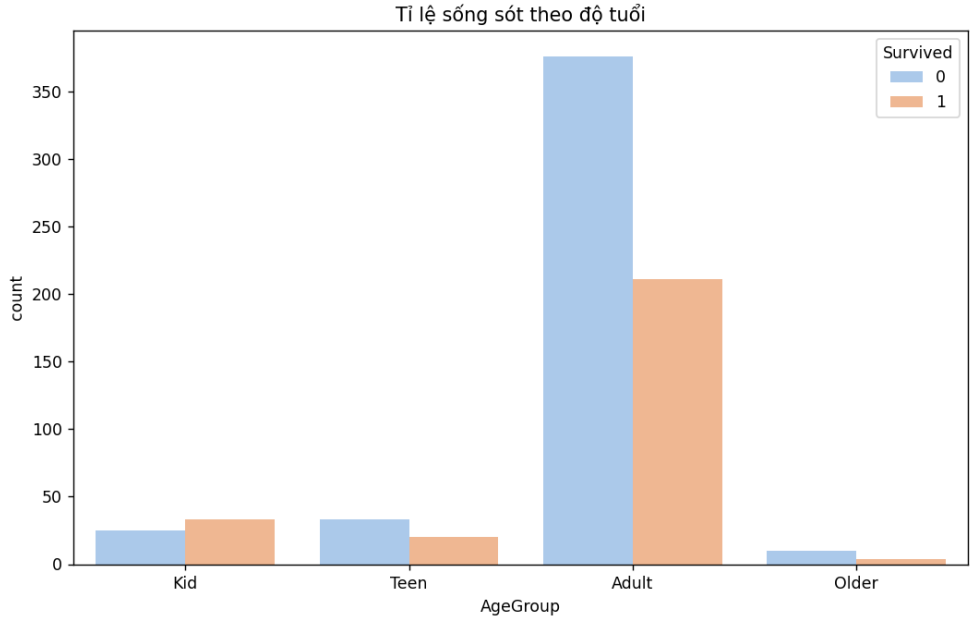
13. Trực quan thông tin hành khách sống sót trên từng nhóm phân loại hành khách (Pclass).

```
# Bài 13
print("Bài 13")

plt.figure(figsize=(8, 5))
sns.countplot(x='Pclass', hue='Survived', data=train, palette='Set3')
plt.title('Tỉ lệ sống sót theo hạng vé (Pclass)')
plt.show()

# Nhận xét:
# - Hạng vé cao (Pclass 1) có tỉ lệ sống sót cao nhất.
# - Hạng vé 3 (giá rẻ) có tỉ lệ tử vong cao nhất.
```



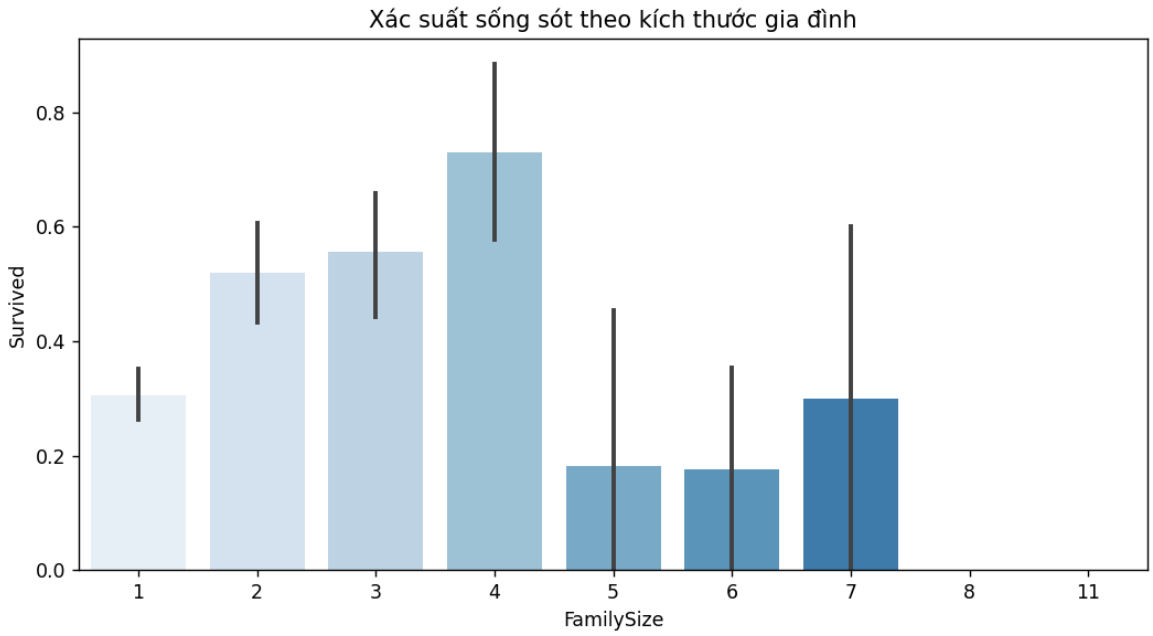
14. Trực quan thông tin hành khách sống sót trên từng nhóm giới tính và thang đo tuổi tác

```
# Bài 14
print("Bài 14")

plt.figure(figsize=(10, 6))
sns.countplot(x='AgeGroup', hue='Survived', data=train, palette='pastel', order=['Kid', 'Teen', 'Adult', 'Older'])
plt.title('Tỉ lệ sống sót theo độ tuổi')
plt.show()

# Nhận xét:
# - Trẻ em (Kid) có tỉ lệ sống sót cao.
# - Người lớn tuổi (Older) tỉ lệ sống sót thấp nhất.
```



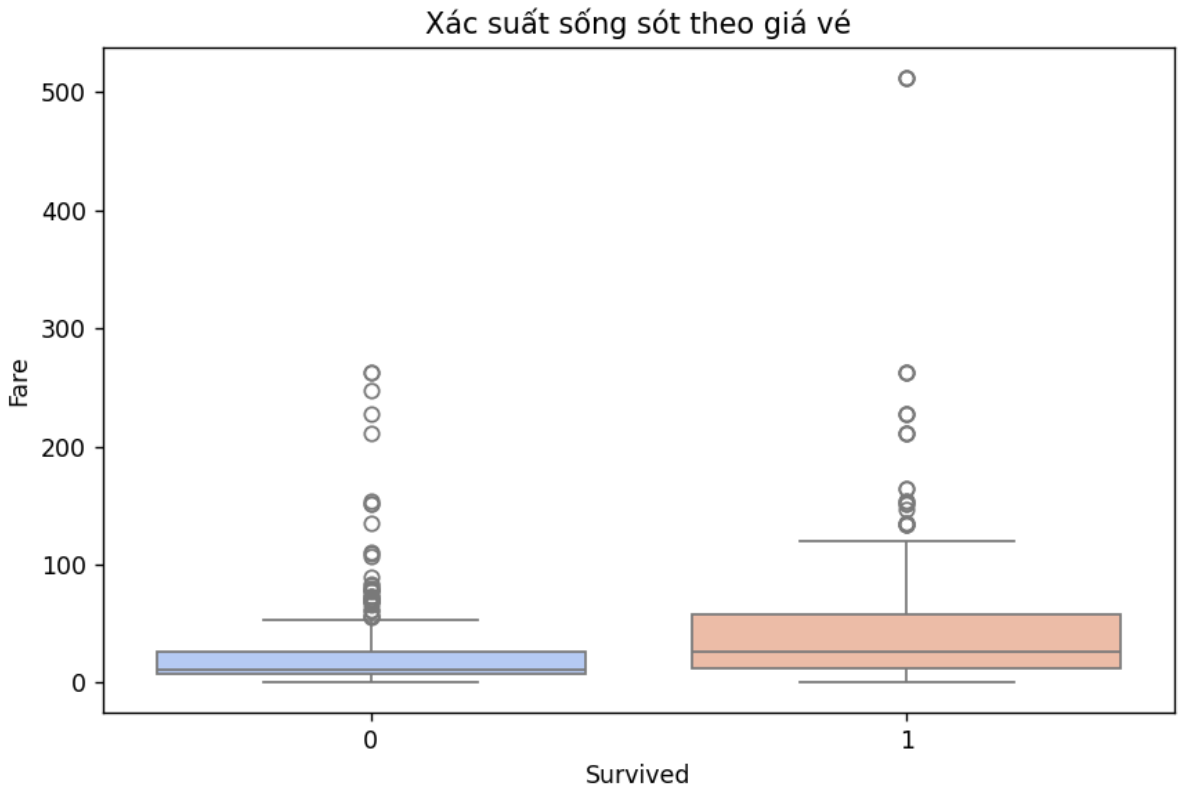
15. Trực quan xác suất hành khách sống sót dựa trên thông tin nhóm đi cùng

```
# Bài 15
print("Bài 15")

plt.figure(figsize=(10, 5))
sns.barplot(x='FamilySize', y='Survived', data=train, palette='Blues')
plt.title('Xác suất sống sót theo kích thước gia đình')
plt.show()

# Nhận xét:
# - Người đi 1 mình (Alone) có tỉ lệ sống sót thấp.
# - Người đi theo nhóm nhỏ (2-4 người) có tỉ lệ sống sót cao nhất.
# - Gia đình quá đông (trên 5 người) tỉ lệ sống sót lại giảm.
```



16. Trực quan xác suất hành khách sống sót dựa trên thông tin giá vé

```
# Bài 16
print("Bài 16")

plt.figure(figsize=(8, 5))
sns.boxplot(x='Survived', y='Fare', data=train, palette='coolwarm')
plt.title('Xác suất sống sót theo giá vé')
plt.show()

# Nhận xét:
# - Giá vé càng cao thì khả năng sống sót càng cao.
# - Điều này liên quan tới việc hạng vé cao (Pclass 1) thường đắt tiền và ưu tiên cứu trước.
```



17. Trực quan số lượng người thiệt mạng và sống sót theo phân lớp (Pclass) hành khách và cảng sẽ cập bến.

```python
# Bài 17
print("Bài 17")

plt.figure(figsize=(10, 6))
sns.catplot(x='Embarked', hue='Survived', col='Pclass', data=train, kind='count', palette='Set1')
plt.suptitle('Tỉ lệ sống sót theo cảng lên tàu và hạng vé')
plt.show()

# Nhận xét:
# - Cảng C (Cherbourg) có tỉ lệ sống sót cao nhất, đặc biệt ở Pclass 1.
# - Cảng S (Southampton) có nhiều hành khách hạng 3 nhất và tỉ lệ tử vong cao.
```