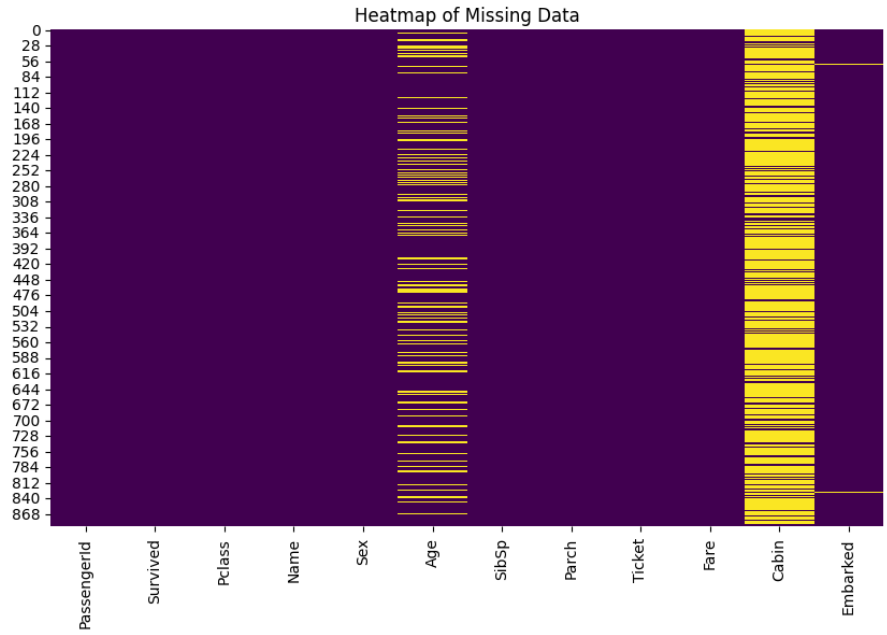
```
import numpy as np
import pandas as pd


#1. Viết hàm load_data() để tải dữ liệu lên ứng dụng. Sau đó, hiển thị ra màn hình 10 dòng đầu tiên
def load_data(filepath):
    data = pd.read_csv(filepath)
    print("10 dòng đầu tiên của dữ liệu:")
    print(data.head(10))
    return data

# Sử dụng hàm để tải dữ liệu
filepath = '/content/titanic_disaster.csv'  # Thay đổi đường dẫn tới file dữ liệu của bạn
data = load_data(filepath)
```

```
10 dòng đầu tiên của dữ liệu:
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3
5            6         0       3
6            7         0       1
7            8         0       3
8            9         1       3
9           10         1       2

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0
5                                   Moran, Mr. James    male   NaN      0
6                            McCarthy, Mr. Timothy J    male  54.0      0
7                     Palsson, Master. Gosta Leonard    male   2.0      3
8  Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)  female  27.0      0
9                Nasser, Mrs. Nicholas (Adele Achem)  female  14.0      1

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
5      0            330877   8.4583   NaN        Q
6      0             17463  51.8625   E46        S
7      1            349909  21.0750   NaN        S
8      2            347742  11.1333   NaN        S
9      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
import seaborn as sns
import matplotlib.pyplot as plt

# Thống kê dữ liệu thiếu
missing_data = data.isnull().sum()
print("Số lượng giá trị thiếu trên mỗi cột:")
print(missing_data)

# Trực quan hóa dữ liệu thiếu
plt.figure(figsize=(10, 6))
sns.heatmap(data.isnull(), cbar=False, cmap='viridis')
plt.title('Heatmap of Missing Data')
plt.show()
```

```
Số lượng giá trị thiếu trên mỗi cột:
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
```




```
column_names=["PassengerId", "Survived", "Pclass", "Name", "Sex", "Age","SibSp", "Parch", "Ticket", "Fare","Cabin","Embarked"]
patientheartrate = pd.read_csv("/content/titanic_disaster.csv", names = column_names)
print(patientheartrate.head())
```

```
   PassengerId  Survived  Pclass  \
0  PassengerId  Survived  Pclass
1            1         0       3
2            2         1       1
3            3         1       3
4            4         1       1

                                                Name     Sex  Age  SibSp  \
0                                               Name     Sex  Age  SibSp
1                            Braund, Mr. Owen Harris    male   22      1
2  Cumings, Mrs. John Bradley (Florence Briggs Th...  female   38      1
3                             Heikkinen, Miss. Laina  female   26      0
4       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female   35      1

   Parch            Ticket     Fare  Cabin  Embarked
0  Parch            Ticket     Fare  Cabin  Embarked
1      0         A/5 21171     7.25    NaN         S
2      0          PC 17599  71.2833    C85         C
3      0  STON/O2. 3101282    7.925    NaN         S
4      0            113803     53.1   C123         S
```

```
#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
patientheartrate[['firstName', 'secondName']] = patientheartrate['Name'].str.split(',', n=1, expand=True)

# Drop the original 'Name' column
patientheartrate = patientheartrate.drop('Name', axis=1)

print(patientheartrate)
```

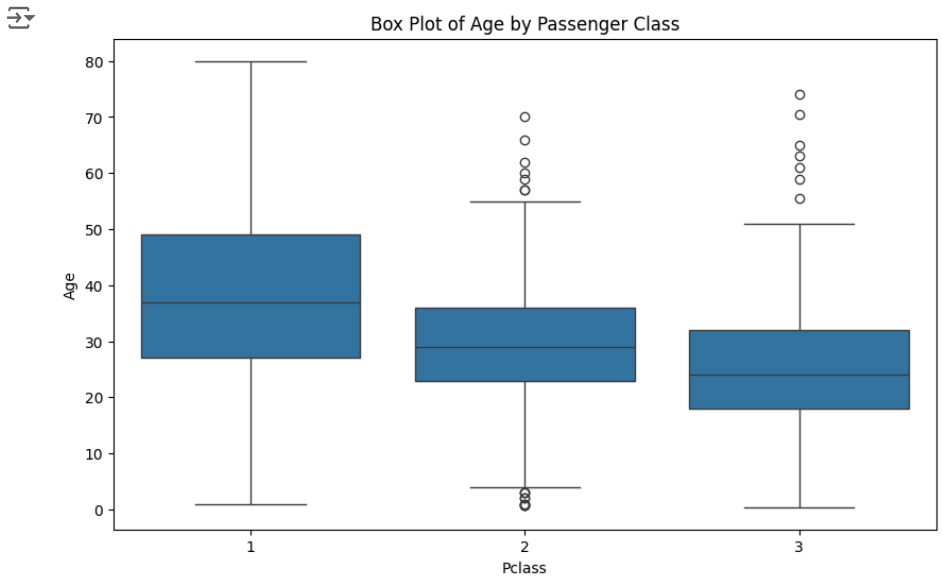
```
     PassengerId  Survived  Pclass     Sex  Age  SibSp  Parch  \
0    PassengerId  Survived  Pclass     Sex  Age  SibSp  Parch
1              1         0       3    male   22      1      0
2              2         1       1  female   38      1      0
3              3         1       3  female   26      0      0
4              4         1       1  female   35      1      0
..           ...       ...     ...     ...  ...    ...    ...
887          887         0       2    male   27      0      0
888          888         1       1  female   19      0      0
889          889         0       3  female  NaN      1      2
890          890         1       1    male   26      0      0
891          891         0       3    male   32      0      0

                Ticket     Fare  Cabin  Embarked  firstName  \
0              Ticket     Fare  Cabin  Embarked       Name
1           A/5 21171     7.25    NaN         S     Braund
2            PC 17599  71.2833    C85         C    Cumings
3    STON/O2. 3101282    7.925    NaN         S  Heikkinen
4              113803     53.1   C123         S   Futrelle
..                ...      ...    ...       ...        ...
887            211536       13    NaN         S   Montvila
888            112053       30    B42         S     Graham
889        W./C. 6607    23.45    NaN         S   Johnston
890            111369       30   C148         C       Behr
891            370376     7.75    NaN         Q     Dooley

                                    secondName
0                                         None
1                              Mr. Owen Harris
2     Mrs. John Bradley (Florence Briggs Thayer)
3                                  Miss. Laina
4             Mrs. Jacques Heath (Lily May Peel)
..                                           ...
887                               Rev. Juozas
888                      Miss. Margaret Edith
889              Miss. Catherine Helen "Carrie"
890                           Mr. Karl Howell
891                               Mr. Patrick

[892 rows x 13 columns]
```

```
#4. Xử lý rút gọn kích thước dữ liệu trên cột Sex như sau: thay thế male  M và female  F
patientheartrate['Sex'] = patientheartrate['Sex'].replace({'male': 'M', 'female': 'F'})
print("10 dòng đầu tiên sau khi xử lý cột Sex:")
print(data.head(10))
```

```
10 dòng đầu tiên sau khi xử lý cột Sex:
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3
5            6         0       3
6            7         0       1
7            8         0       3
8            9         1       3
9           10         1       2

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0
5                                   Moran, Mr. James    male   NaN      0
6                            McCarthy, Mr. Timothy J    male  54.0      0
7                     Palsson, Master. Gosta Leonard    male   2.0      3
8  Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)  female  27.0      0
9                Nasser, Mrs. Nicholas (Adele Achem)  female  14.0      1

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
```

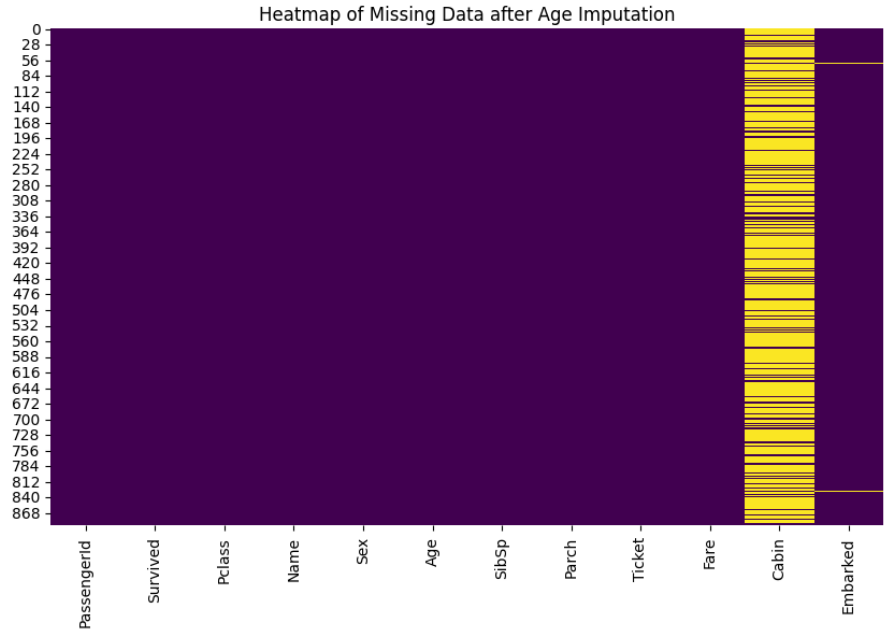
```
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
5      0            330877   8.4583   NaN        Q
6      0             17463  51.8625   E46        S
7      1            349909  21.0750   NaN        S
8      2            347742  11.1333   NaN        S
9      0            237736  30.0708   NaN        C
```

```
#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 t
#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

plt.figure(figsize=(10, 6))
sns.boxplot(x='Pclass', y='Age', data=data)
plt.title('Box Plot of Age by Passenger Class')
plt.show()
```




```
#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
# Tính giá trị trung bình của tuổi theo từng nhóm hạng vé
mean_age_by_pclass = data.groupby('Pclass')['Age'].mean()
print("Giá trị trung bình của tuổi theo từng nhóm hạng vé:")
print(mean_age_by_pclass)

# Thay thế giá trị thiếu
def fill_age(row):
    if pd.isnull(row['Age']):
        return mean_age_by_pclass[row['Pclass']]
    else:
        return row['Age']

data['Age'] = data.apply(fill_age, axis=1)

# Kiểm tra lại dữ liệu sau khi xử lý thiếu
missing_data_after = data.isnull().sum()
print("Số lượng giá trị thiếu trên mỗi cột sau khi xử lý:")
print(missing_data_after)

# Trực quan hóa lại dữ liệu thiếu
plt.figure(figsize=(10, 6))
sns.heatmap(data.isnull(), cbar=False, cmap='viridis')
plt.title('Heatmap of Missing Data after Age Imputation')
plt.show()
```

```
Giá trị trung bình của tuổi theo từng nhóm hạng vé:
Pclass
1    38.233441
2    29.877630
3    25.140620
Name: Age, dtype: float64
Số lượng giá trị thiếu trên mỗi cột sau khi xử lý:
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```



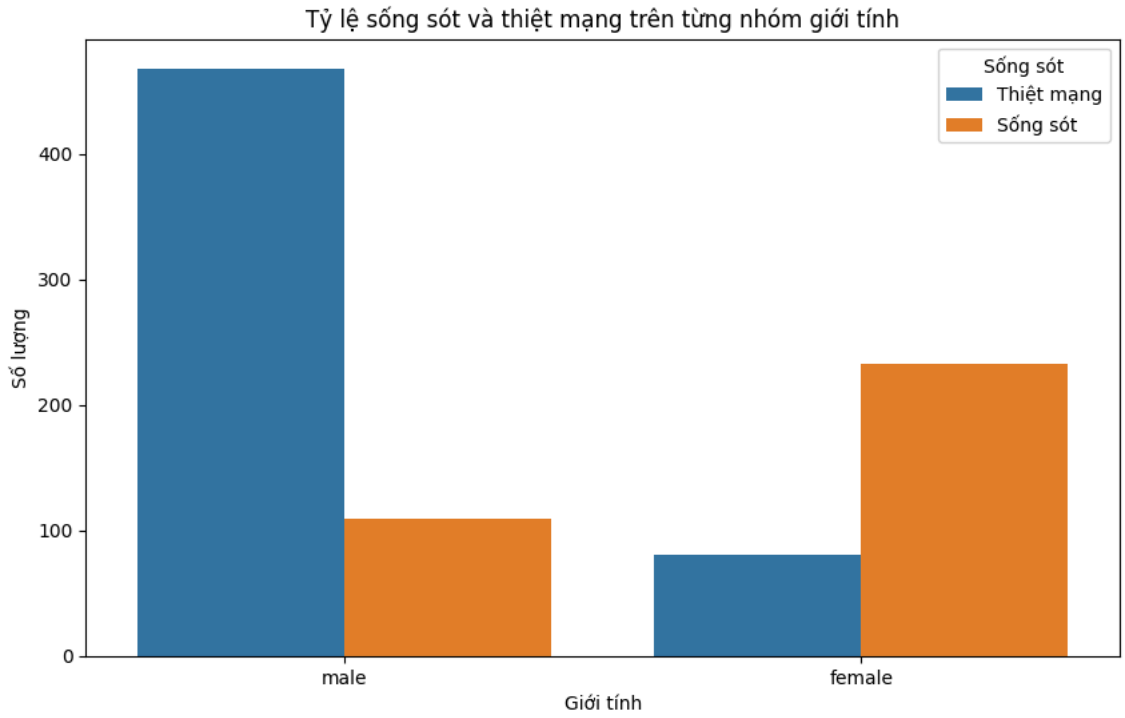
#6. Xây dựng biến số Agegroup có thang đo thứ tự được ánh xạ theo thang đo khoảng dựa trên độ tuổi của hành khách như sau: (age =< 12] ⍰ Ki

```
import seaborn as sns
import matplotlib.pyplot as plt
titanic = sns.load_dataset('titanic')
def age_group(age):
    if age <= 12:
        return 'Kid'
    elif age <= 18:
        return 'Teen'
    elif age <= 60:
        return 'Adult'
    else:
        return 'Older'

titanic['age_group'] = titanic['age'].apply(age_group)
```

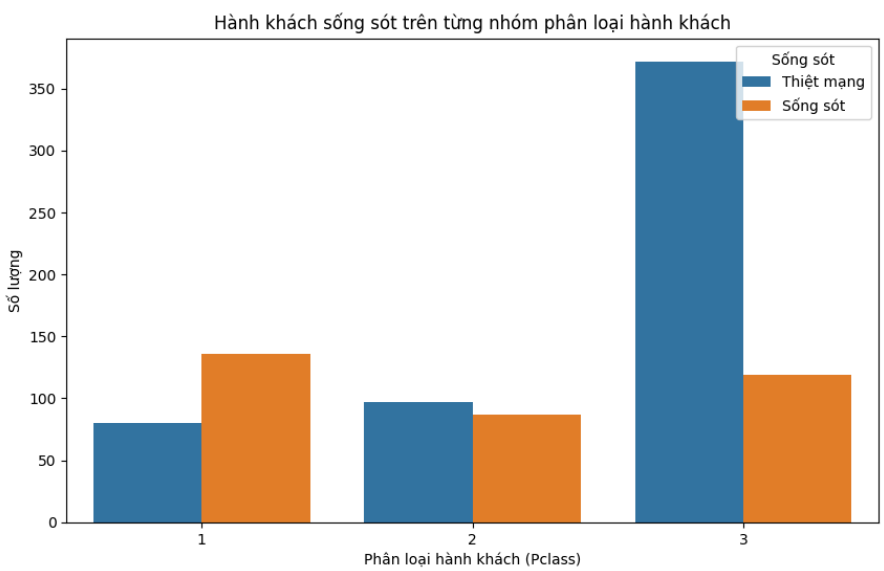
```
#7. Tiến hành thêm đặc trưng về danh xưng (namePrefix) trong xã hội bằng cách tách Mr, Mrs, Miss, Master ra khỏi –secondName
# 7. Thêm đặc trưng về danh xưng (namePrefix)
def extract_prefix(name):
    if pd.isnull(name):
        return 'Unknown'
    elif ',' in name:
        return name.split(', ')[1].split('.')[0]
    else:
        return 'No Prefix' # Handle cases without commas

titanic['namePrefix'] = titanic['who'].apply(extract_prefix)


#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 s
titanic['family_size'] = 1 + titanic['sibsp'] + titanic['parch']


#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 th
titanic['alone'] = titanic['family_size'].apply(lambda x: 1 if x == 1 else 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.
def extract_cabin_type(cabin):
    if pd.isnull(cabin):
        return 'Unknown'
    return cabin[0]

# Verify if 'cabin' column exists. If not, it might be 'Cabin'
if 'cabin' not in titanic.columns:
    if 'Cabin' in titanic.columns:
        print("Column name is 'Cabin', not 'cabin'")
        titanic['typeCabin'] = titanic['Cabin'].apply(extract_cabin_type)
    else:
        print("Neither 'cabin' nor 'Cabin' column exists. Check previous steps.")
else:
    titanic['typeCabin'] = titanic['cabin'].apply(extract_cabin_type)
```

```
Neither 'cabin' nor 'Cabin' column exists. Check previous steps.
```

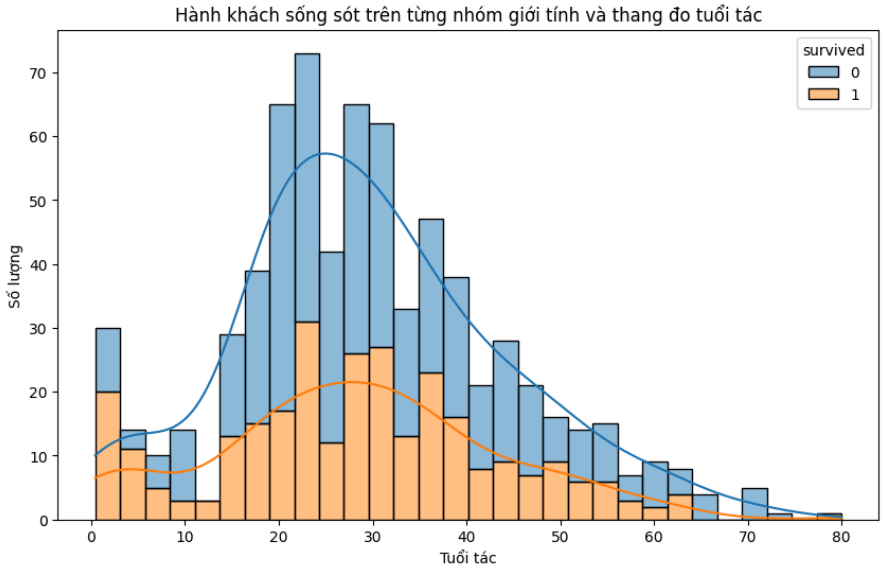
```
#12. Trực quan thông tin tương quan tỉ lệ sống sót và thiệt mạng trên từng nhóm giới tính.
plt.figure(figsize=(10, 6))
sns.countplot(x='sex', hue='survived', data=titanic)
plt.title('Tỷ lệ sống sót và thiệt mạng trên từng nhóm giới tính')
plt.xlabel('Giới tính')
plt.ylabel('Số lượng')
plt.legend(title='Sống sót', loc='upper right', labels=['Thiệt mạng', 'Sống sót'])
plt.show()
```



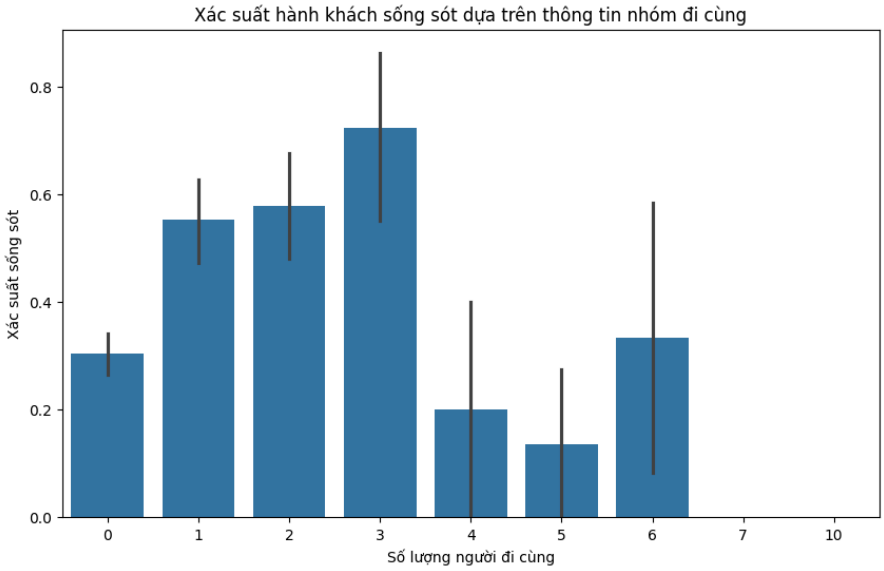
```
#13. Trực quan thông tin hành khách sống sót trên từng nhóm phân loại hành khách (Pclass).
plt.figure(figsize=(10, 6))
sns.countplot(x='pclass', hue='survived', data=titanic)
plt.title('Hành khách sống sót trên từng nhóm phân loại hành khách')
plt.xlabel('Phân loại hành khách (Pclass)')
plt.ylabel('Số lượng')
plt.legend(title='Sống sót', loc='upper right', labels=['Thiệt mạng', 'Sống sót'])
plt.show()
```



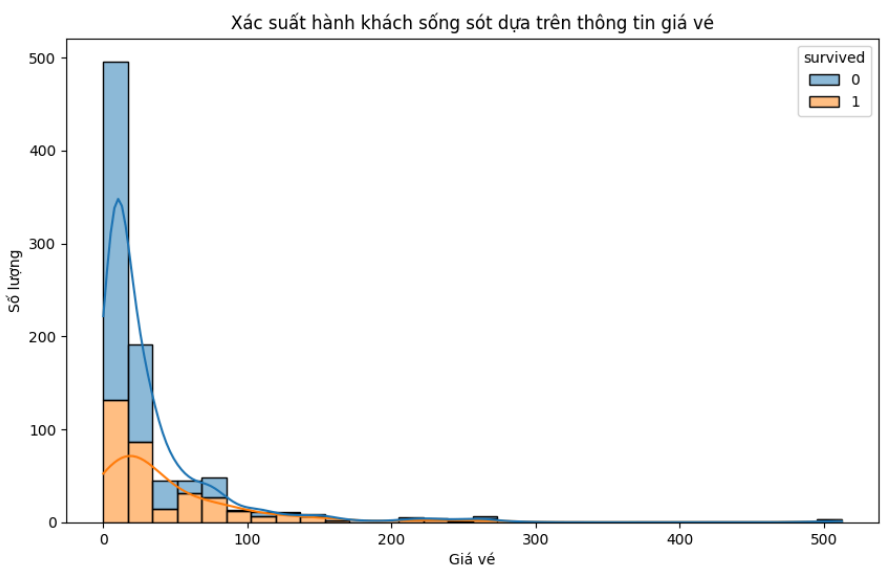

```
#14. Trực quan thông tin hành khách sống sót trên từng nhóm giới tính và thang đo tuổi tác
plt.figure(figsize=(10, 6))
sns.histplot(data=titanic, x='age', hue='survived', multiple='stack', kde=True, bins=30)
plt.title('Hành khách sống sót trên từng nhóm giới tính và thang đo tuổi tác')
plt.xlabel('Tuổi tác')
plt.ylabel('Số lượng')
plt.show()
```




```
#15. Trực quan xác suất hành khách sống sót dựa trên thông tin nhóm đi cùng
titanic['family_size'] = titanic['sibsp'] + titanic['parch']
plt.figure(figsize=(10, 6))
sns.barplot(x='family_size', y='survived', data=titanic)
plt.title('Xác suất hành khách sống sót dựa trên thông tin nhóm đi cùng')
plt.xlabel('Số lượng người đi cùng')
plt.ylabel('Xác suất sống sót')
plt.show()
```



```
#16. Trực quan xác suất hành khách sống sót dựa trên thông tin giá vé
plt.figure(figsize=(10, 6))
sns.histplot(data=titanic, x='fare', hue='survived', multiple='stack', kde=True, bins=30)
plt.title('Xác suất hành khách sống sót dựa trên thông tin giá vé')
plt.xlabel('Giá vé')
plt.ylabel('Số lượng')
plt.show()
```




```
#17. Trực quan số lượng người thiệt mạng và sống sót theo phân lớp (Pclass) hành khách và cảng sẽ cập bến.
plt.figure(figsize=(10, 6))
```

Không thể kết nối với dịch vụ reCAPTCHA. Vui lòng kiểm tra kết nối internet của bạn và tải lại để nhận hình ảnh xác thực reCAPTCHA.