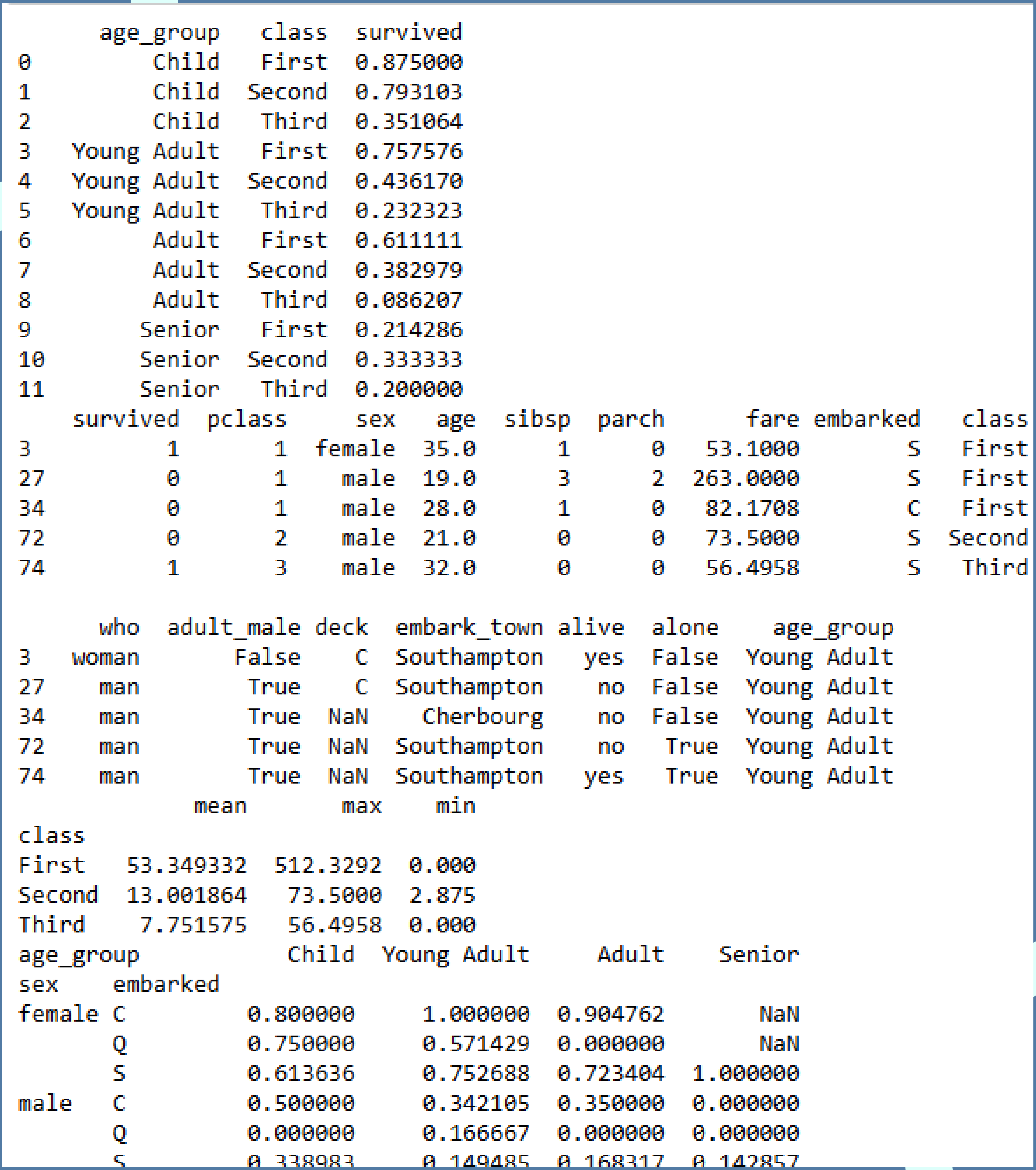
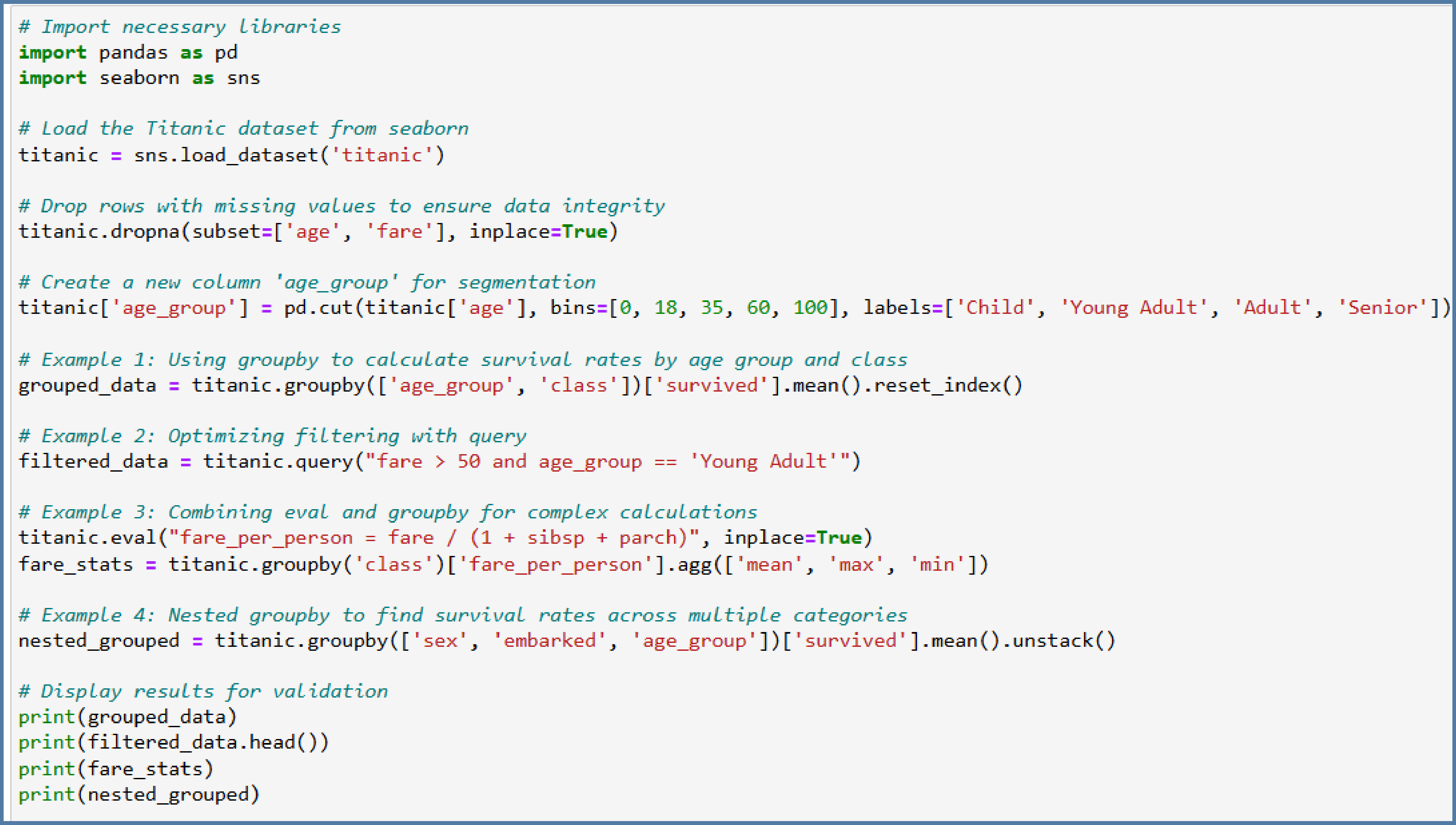


```
      age_group   class  survived
0         Child   First  0.875000
1         Child  Second  0.793103
2         Child   Third  0.351064
3   Young Adult   First  0.757576
4   Young Adult  Second  0.436170
5   Young Adult   Third  0.232323
6         Adult   First  0.611111
7         Adult  Second  0.382979
8         Adult   Third  0.086207
9        Senior   First  0.214286
10       Senior  Second  0.333333
11       Senior   Third  0.200000
    survived  pclass     sex   age  sibsp  parch      fare embarked   class
3          1       1  female  35.0      1      0   53.1000        S   First
27         0       1    male  19.0      3      2  263.0000        S   First
34         0       1    male  28.0      1      0   82.1708        C   First
72         0       2    male  21.0      0      0   73.5000        S  Second
74         1       3    male  32.0      0      0   56.4958        S   Third

      who  adult_male deck  embark_town alive  alone    age_group
3   woman       False    C  Southampton   yes  False  Young Adult
27    man        True    C  Southampton    no  False  Young Adult
34    man        True  NaN    Cherbourg    no  False  Young Adult
72    man        True  NaN  Southampton    no   True  Young Adult
74    man        True  NaN  Southampton   yes   True  Young Adult
             mean       max    min
class
First   53.349332  512.3292  0.000
Second  13.001864   73.5000  2.875
Third    7.751575   56.4958  0.000
age_group            Child  Young Adult     Adult    Senior
sex    embarked
female C          0.800000     1.000000  0.904762       NaN
       Q          0.750000     0.571429  0.000000       NaN
       S          0.613636     0.752688  0.723404  1.000000
male   C          0.500000     0.342105  0.350000  0.000000
       Q          0.000000     0.166667  0.000000  0.000000
       S          0.338983     0.149485  0.168317  0.142857
```



```
# Import necessary libraries
import pandas as pd
import seaborn as sns

# Load the Titanic dataset from seaborn
titanic = sns.load_dataset('titanic')

# Drop rows with missing values to ensure data integrity
titanic.dropna(subset=['age', 'fare'], inplace=True)

# Create a new column 'age_group' for segmentation
titanic['age_group'] = pd.cut(titanic['age'], bins=[0, 18, 35, 60, 100], labels=['Child', 'Young Adult', 'Adult', 'Senior'])

# Example 1: Using groupby to calculate survival rates by age group and class
grouped_data = titanic.groupby(['age_group', 'class'])['survived'].mean().reset_index()

# Example 2: Optimizing filtering with query
filtered_data = titanic.query("fare > 50 and age_group == 'Young Adult'")

# Example 3: Combining eval and groupby for complex calculations
titanic.eval("fare_per_person = fare / (1 + sibsp + parch)", inplace=True)
fare_stats = titanic.groupby('class')['fare_per_person'].agg(['mean', 'max', 'min'])

# Example 4: Nested groupby to find survival rates across multiple categories
nested_grouped = titanic.groupby(['sex', 'embarked', 'age_group'])['survived'].mean().unstack()

# Display results for validation
print(grouped_data)
print(filtered_data.head())
print(fare_stats)
print(nested_grouped)
```