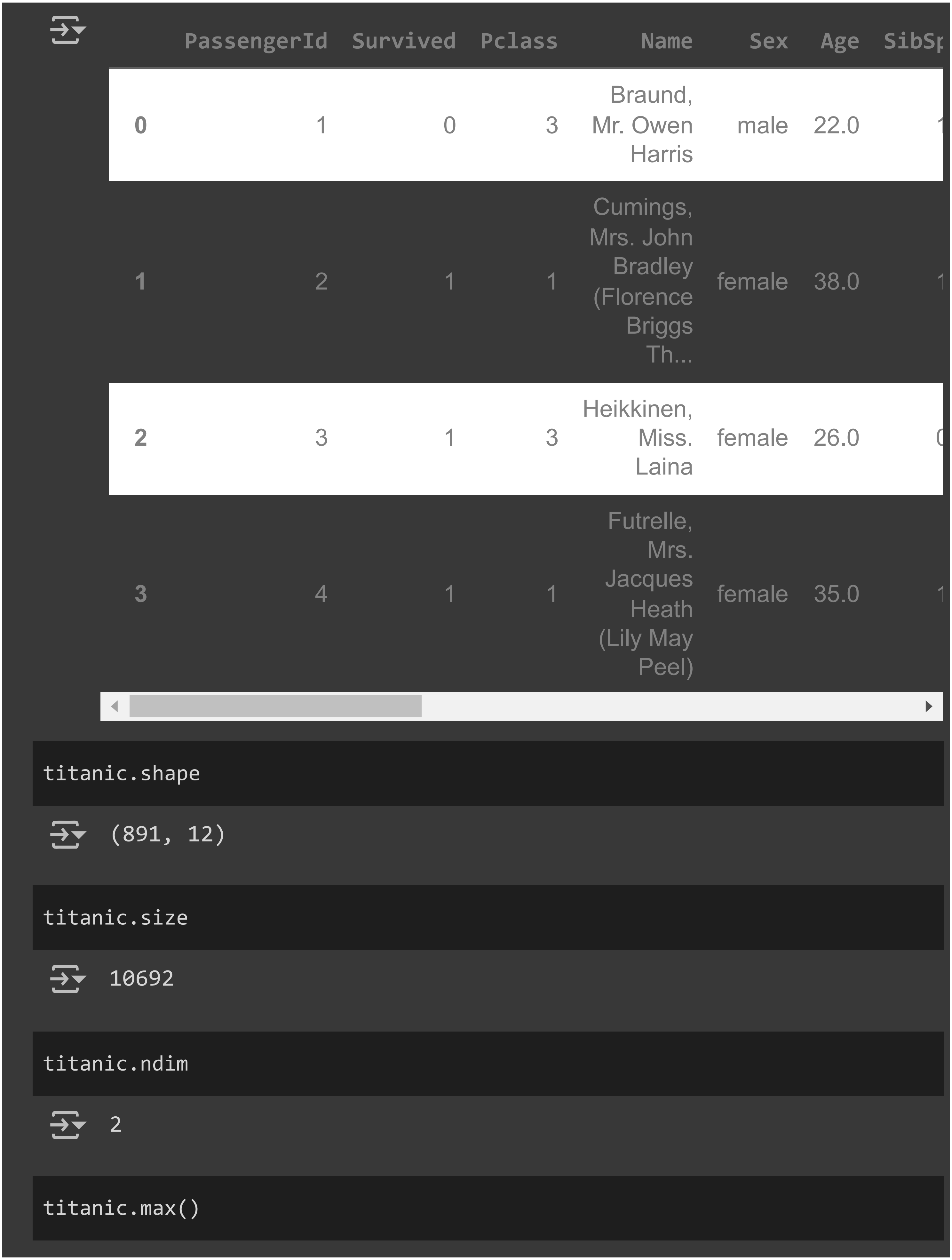
```
# understanding the data and issue and associated with data
# we will  find out how the servival rate
# 1 - servived
# 0 - not servivrd
# we will find out how the servival rate of a person is depending on the pa

## Story Of The DataSet
# clean the data set and data meaning

#  >>>>>>missing data handling<<<<<<<<
#  p class--- passenger class
# 1 upper class
# 2 middle class
# 3 lower class

# SibSp --- silbilg (brother, sister stepbrother, stepsister)
# Spouse==== husband, wife, (mistresses and fiances were ignored)
# s== southampton
#c=== cherbourg
# Q=== queenstown
# embarked== port of embarkation
```

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

```
titanic=pd.read_csv('/content/sample_data/27 titanic.csv')
```

```
titanic
```



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1

```
titanic.shape
```

```
(891, 12)
```

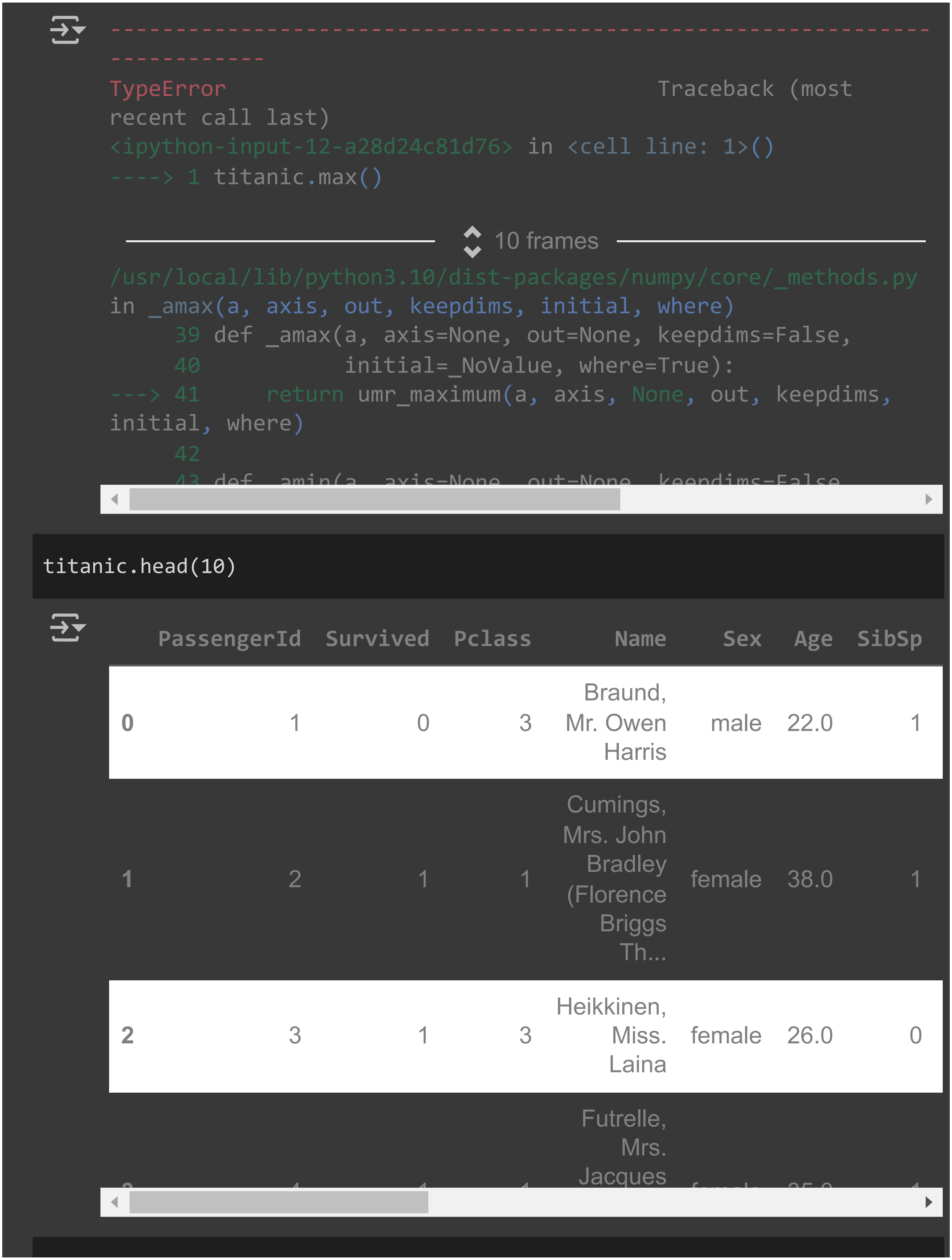
```
titanic.size
```

```
10692
```

```
titanic.ndim
```

```
2
```

```
titanic.max()
```



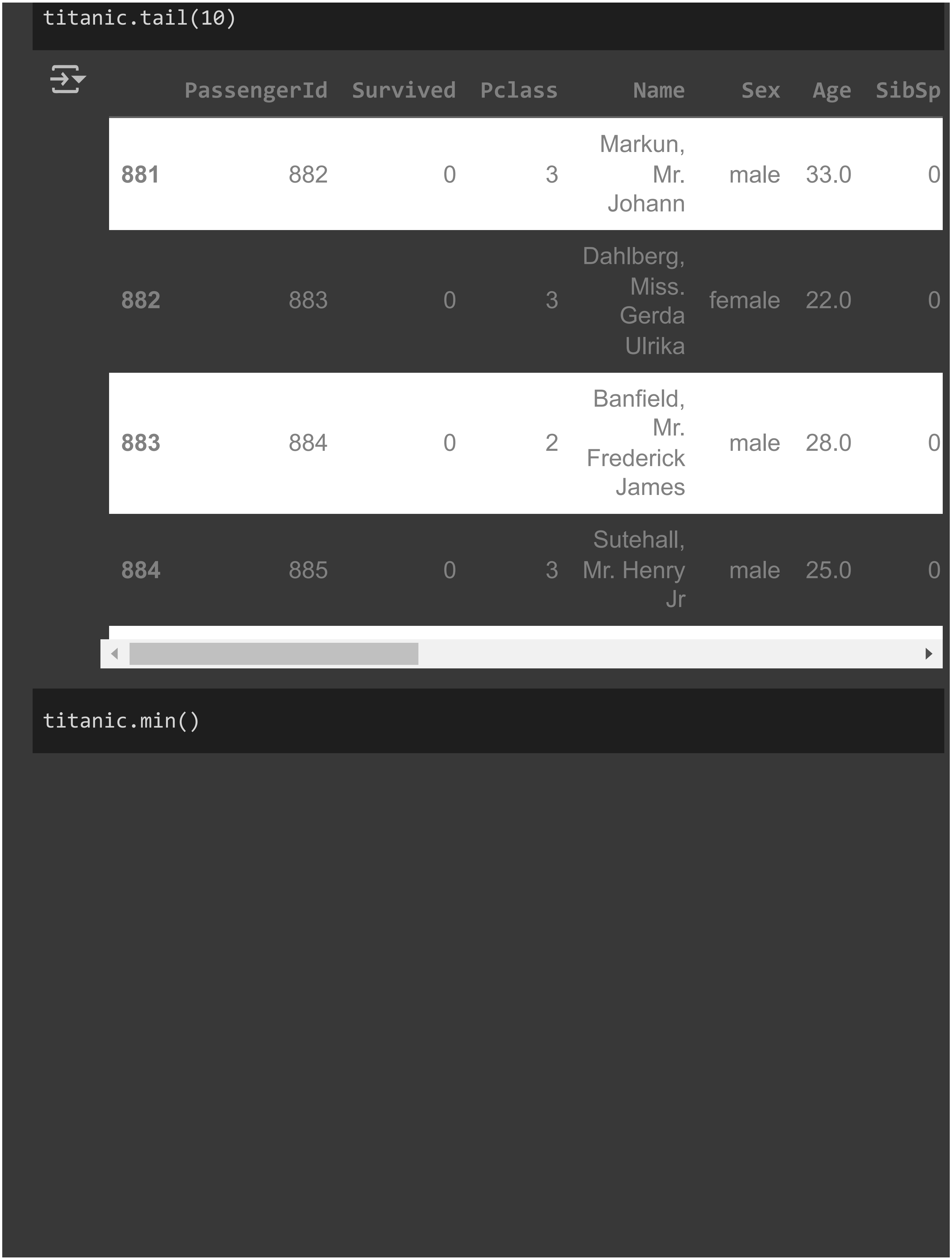
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-12-a28d24c81d76> in <cell line: 1>()
----> 1 titanic.max()
```

10 frames

```
/usr/local/lib/python3.10/dist-packages/numpy/core/_methods.py in _amax(a, axis, out, keepdims, initial, where)
     39 def _amax(a, axis=None, out=None, keepdims=False,
     40           initial=_NoValue, where=True):
---> 41     return umr_maximum(a, axis, None, out, keepdims, initial, where)
     42
     43 def _amin(a, axis=None, out=None, keepdims=False,
```

```
titanic.head(10)
```

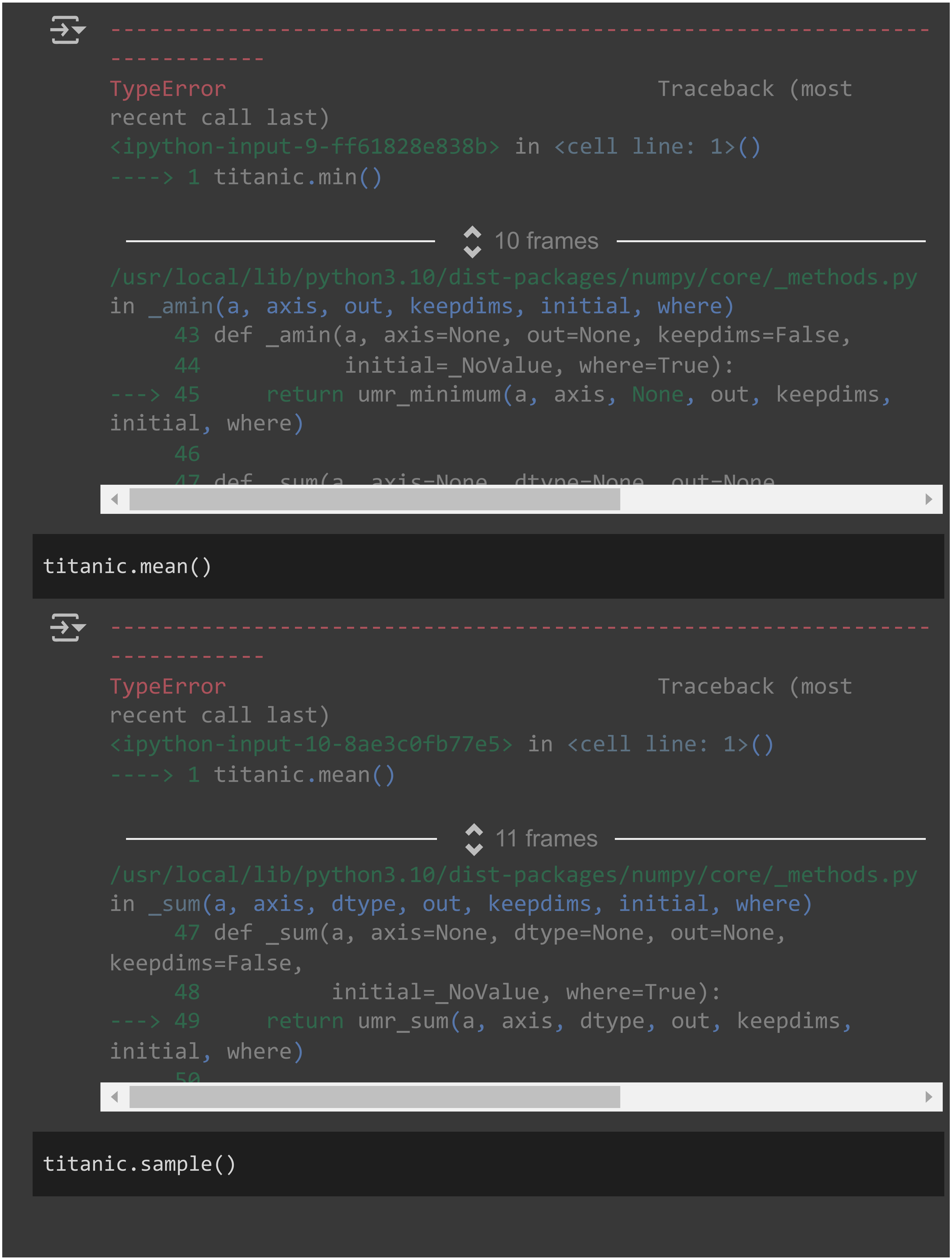
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0
3	4	1	1	Futrelle, Mrs. Jacques	female	35.0	1



```
titanic.tail(10)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp
881	882	0	3	Markun, Mr. Johann	male	33.0	0
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22.0	0
883	884	0	2	Banfield, Mr. Frederick James	male	28.0	0
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0

```
titanic.min()
```



```
-----------------------------------------------------------------------
------------
TypeError                                   Traceback (most
recent call last)
<ipython-input-9-ff61828e838b> in <cell line: 1>()
----> 1 titanic.min()
```

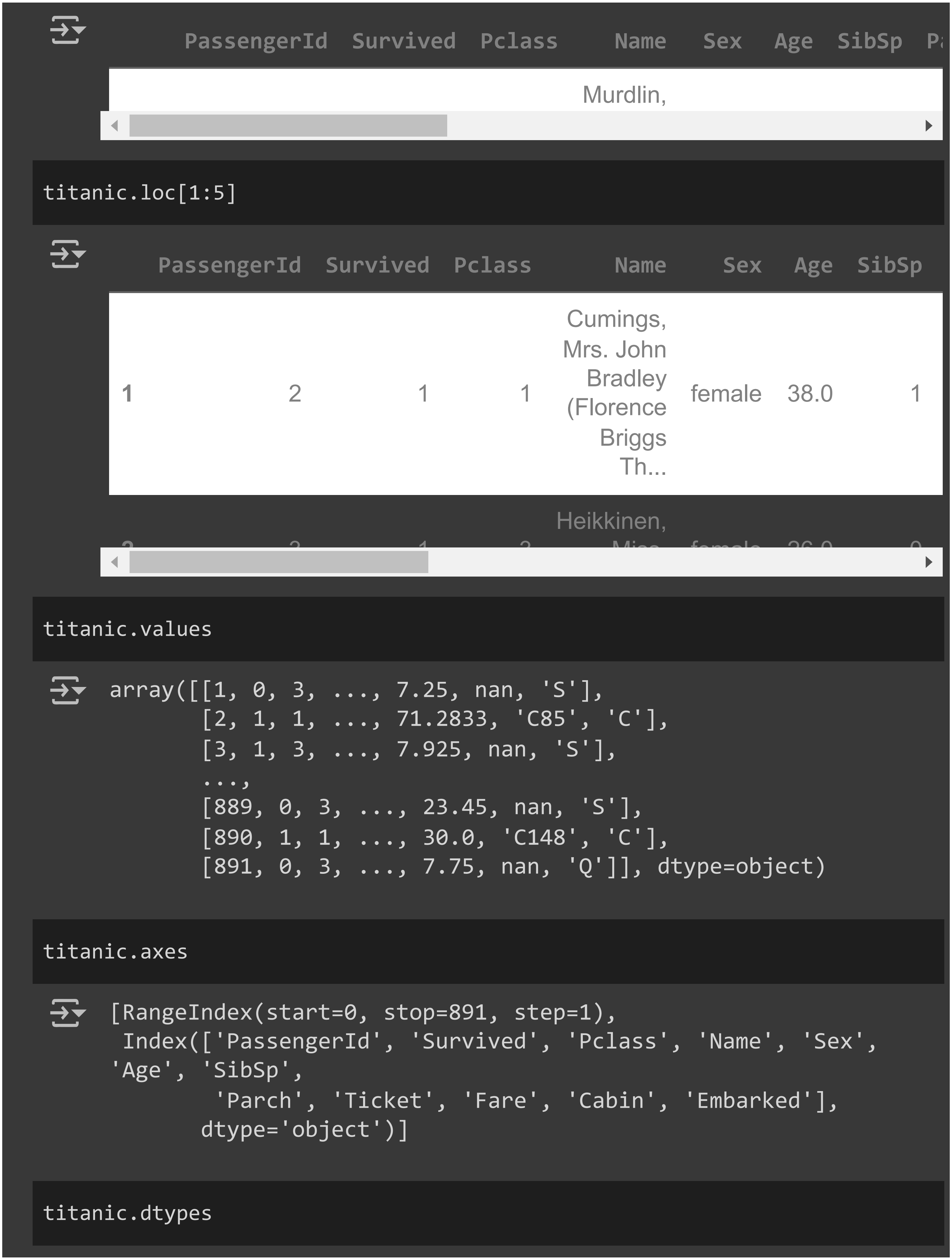
```
/usr/local/lib/python3.10/dist-packages/numpy/core/_methods.py
in _amin(a, axis, out, keepdims, initial, where)
     43 def _amin(a, axis=None, out=None, keepdims=False,
     44           initial=_NoValue, where=True):
---> 45     return umr_minimum(a, axis, None, out, keepdims,
initial, where)
     46
     47 def _sum(a, axis=None, dtype=None, out=None,
```

```
titanic.mean()
```

```
-----------------------------------------------------------------------
------------
TypeError                                   Traceback (most
recent call last)
<ipython-input-10-8ae3c0fb77e5> in <cell line: 1>()
----> 1 titanic.mean()
```

```
/usr/local/lib/python3.10/dist-packages/numpy/core/_methods.py
in _sum(a, axis, dtype, out, keepdims, initial, where)
     47 def _sum(a, axis=None, dtype=None, out=None,
keepdims=False,
     48          initial=_NoValue, where=True):
---> 49     return umr_sum(a, axis, dtype, out, keepdims,
initial, where)
     50
```

```
titanic.sample()
```



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Pa
				Murdlin,				

```
titanic.loc[1:5]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1
2	3	1	3	Heikkinen, Miss	female	26.0	0

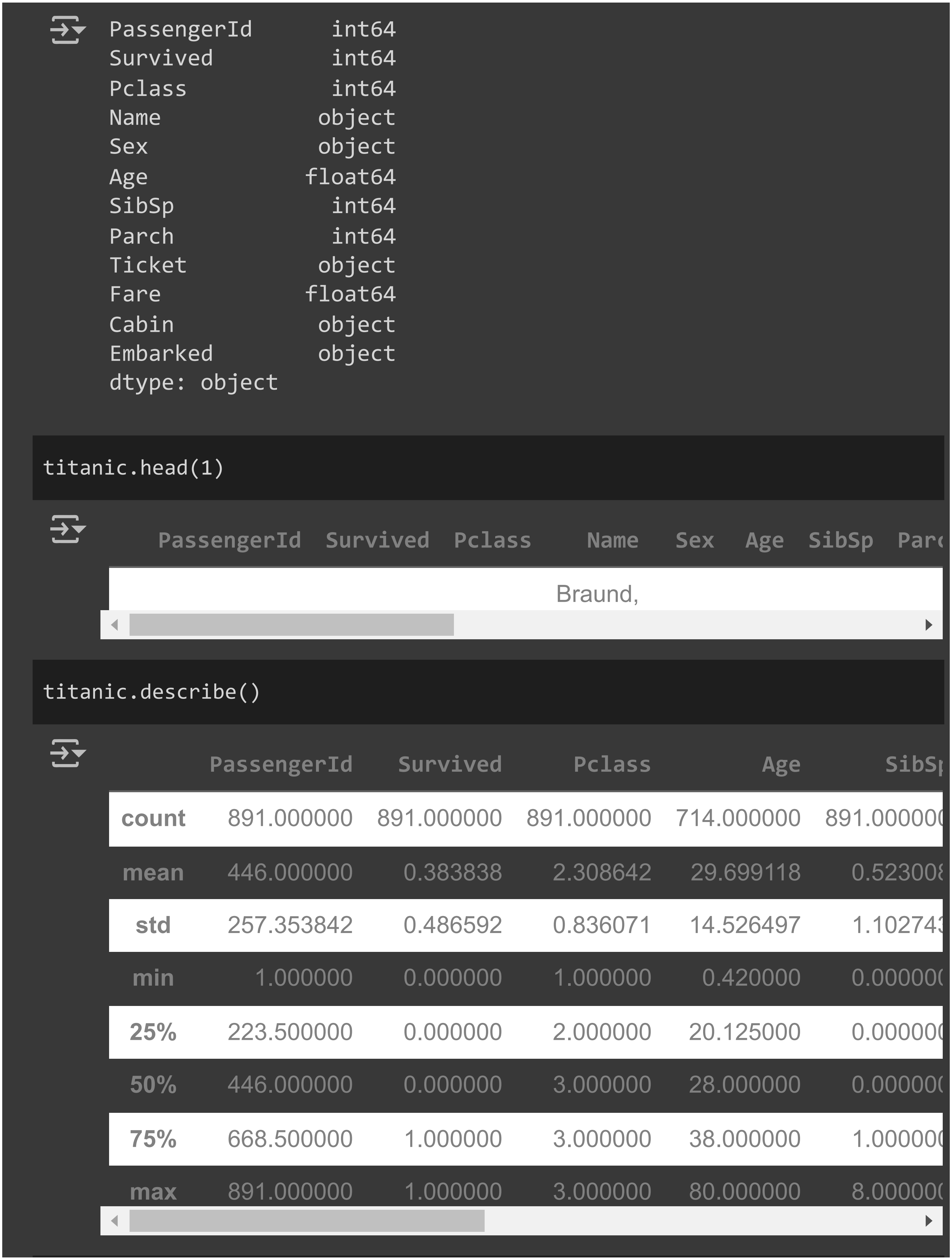
```
titanic.values
```

```
array([[1, 0, 3, ..., 7.25, nan, 'S'],
       [2, 1, 1, ..., 71.2833, 'C85', 'C'],
       [3, 1, 3, ..., 7.925, nan, 'S'],
       ...,
       [889, 0, 3, ..., 23.45, nan, 'S'],
       [890, 1, 1, ..., 30.0, 'C148', 'C'],
       [891, 0, 3, ..., 7.75, nan, 'Q']], dtype=object)
```

```
titanic.axes
```

```
[RangeIndex(start=0, stop=891, step=1),
 Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex',
'Age', 'SibSp',
        'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
       dtype='object')]
```

```
titanic.dtypes
```



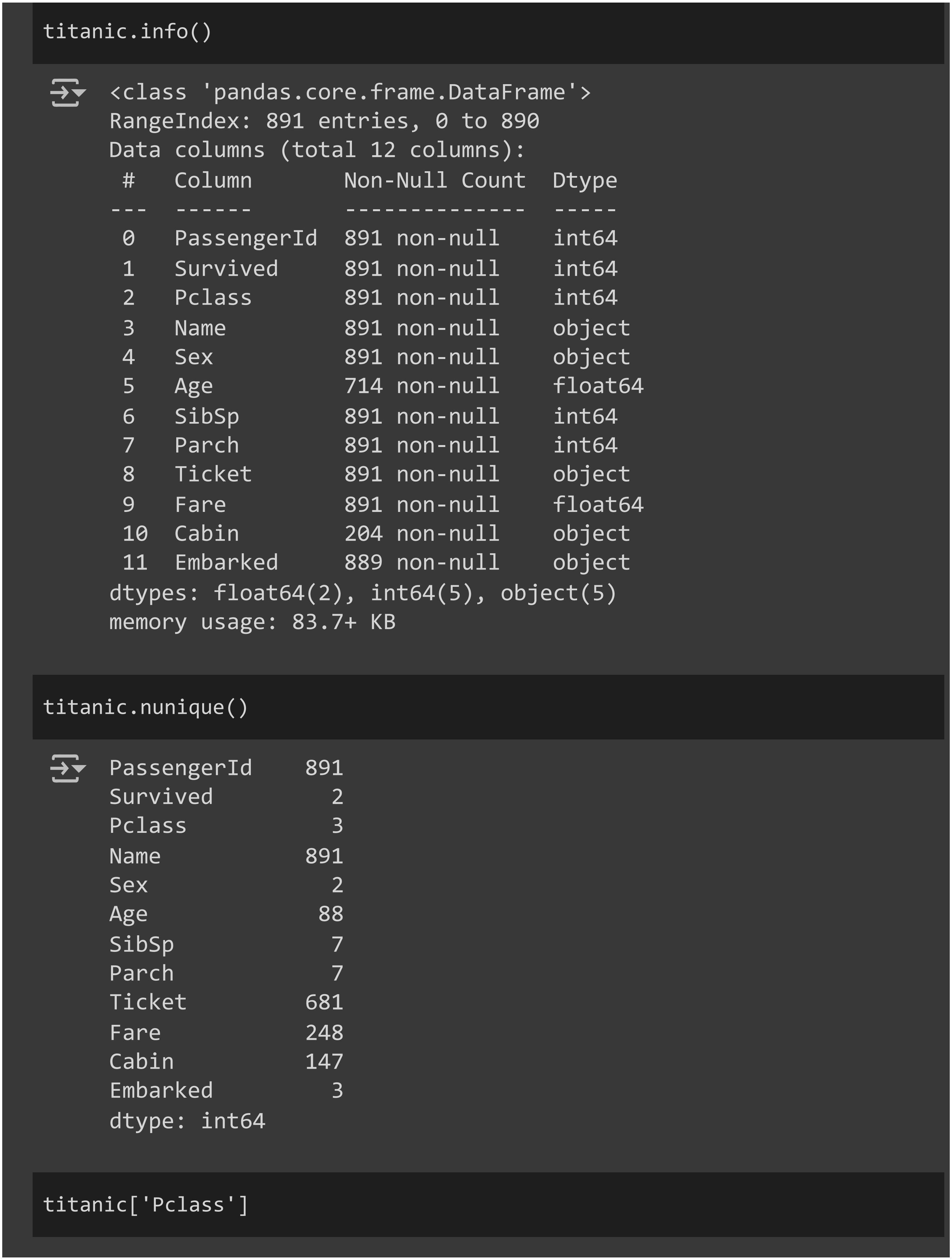
```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

```
titanic.head(1)
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parc
			Braund,				

```
titanic.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp
count	891.000000	891.000000	891.000000	714.000000	891.00000
mean	446.000000	0.383838	2.308642	29.699118	0.52300
std	257.353842	0.486592	0.836071	14.526497	1.10274
min	1.000000	0.000000	1.000000	0.420000	0.00000
25%	223.500000	0.000000	2.000000	20.125000	0.00000
50%	446.000000	0.000000	3.000000	28.000000	0.00000
75%	668.500000	1.000000	3.000000	38.000000	1.00000
max	891.000000	1.000000	3.000000	80.000000	8.00000



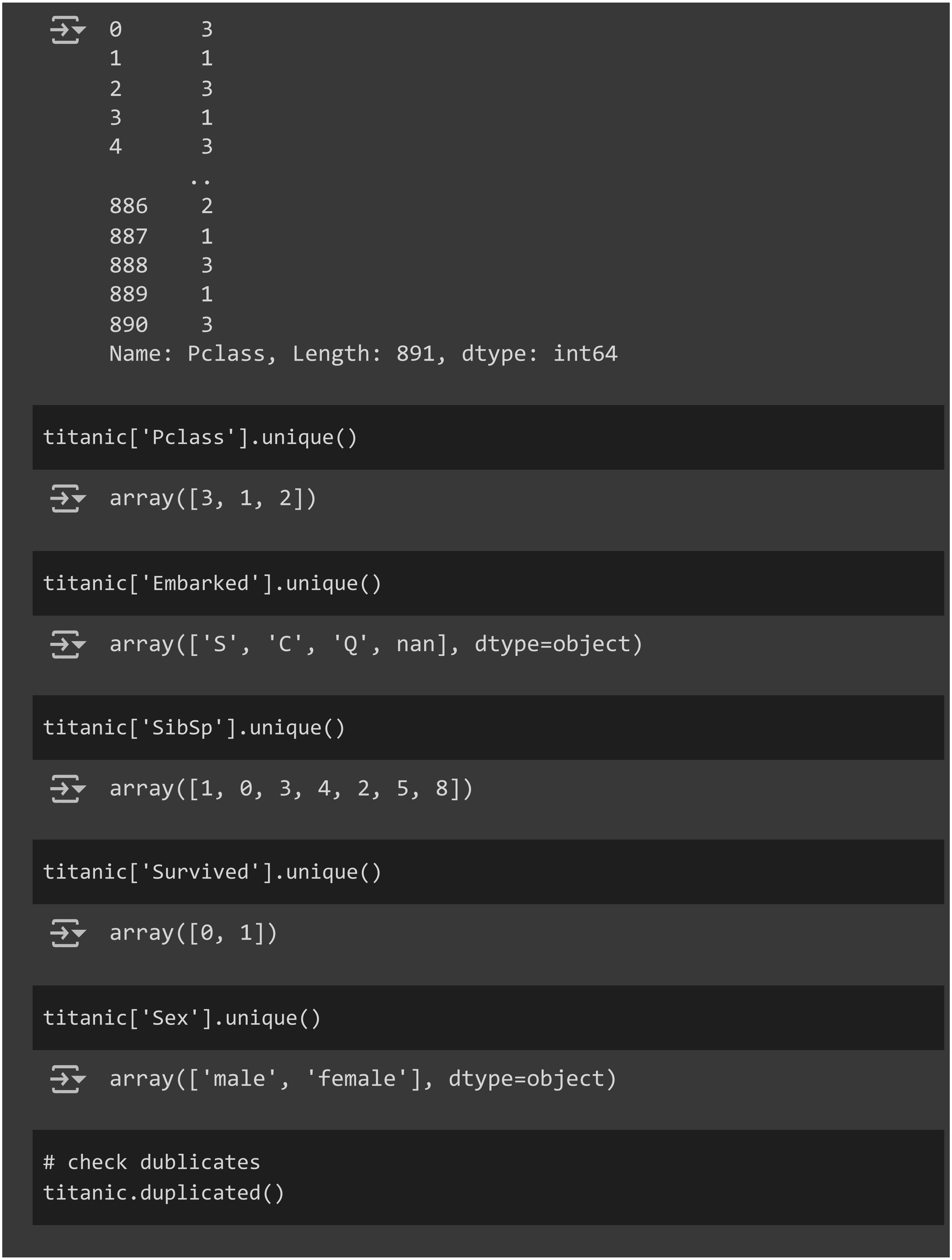
```
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
titanic.nunique()
```

```
PassengerId    891
Survived         2
Pclass           3
Name           891
Sex              2
Age             88
SibSp            7
Parch            7
Ticket         681
Fare           248
Cabin          147
Embarked         3
dtype: int64
```

```
titanic['Pclass']
```



```
0      3
1      1
2      3
3      1
4      3
      ..
886    2
887    1
888    3
889    1
890    3
Name: Pclass, Length: 891, dtype: int64
```

```
titanic['Pclass'].unique()
```

```
array([3, 1, 2])
```

```
titanic['Embarked'].unique()
```

```
array(['S', 'C', 'Q', nan], dtype=object)
```

```
titanic['SibSp'].unique()
```

```
array([1, 0, 3, 4, 2, 5, 8])
```

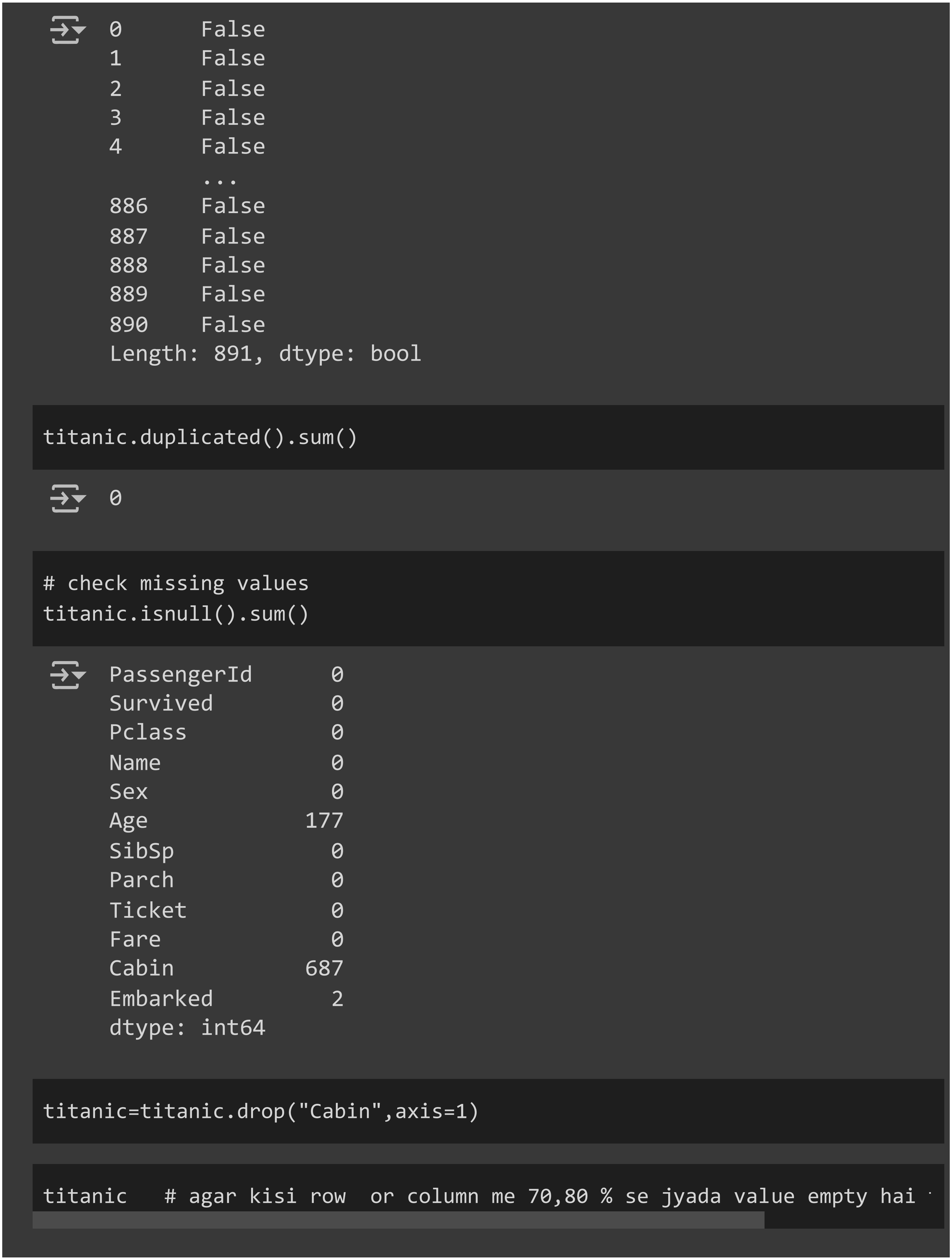
```
titanic['Survived'].unique()
```

```
array([0, 1])
```

```
titanic['Sex'].unique()
```

```
array(['male', 'female'], dtype=object)
```

```
# check dublicates
titanic.duplicated()
```



```
0      False
1      False
2      False
3      False
4      False
       ...
886    False
887    False
888    False
889    False
890    False
Length: 891, dtype: bool
```

```
titanic.duplicated().sum()
```

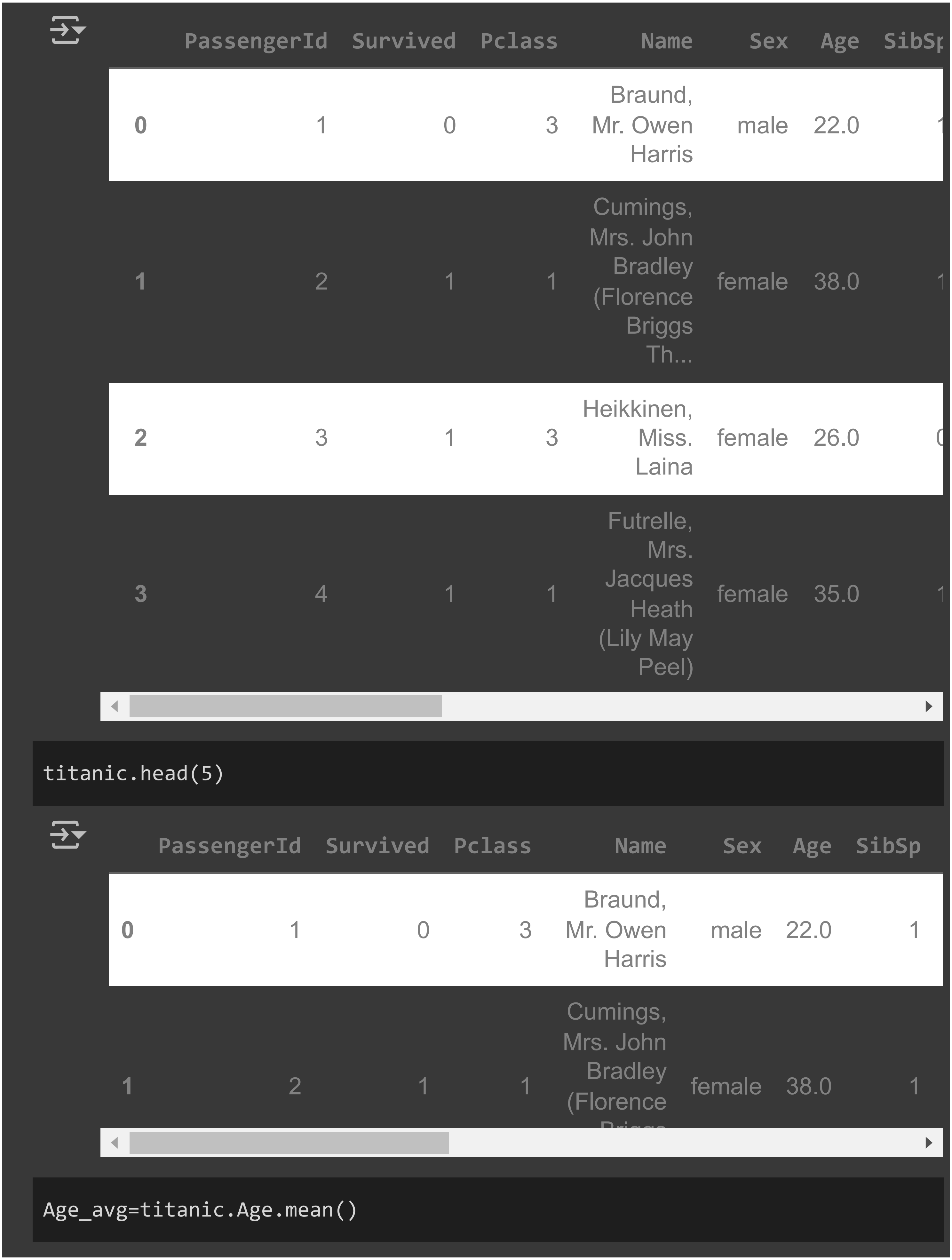
```
0
```

```
# check missing values
titanic.isnull().sum()
```

```
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
```

```
titanic=titanic.drop("Cabin",axis=1)
```

```
titanic   # agar kisi row  or column me 70,80 % se jyada value empty hai
```

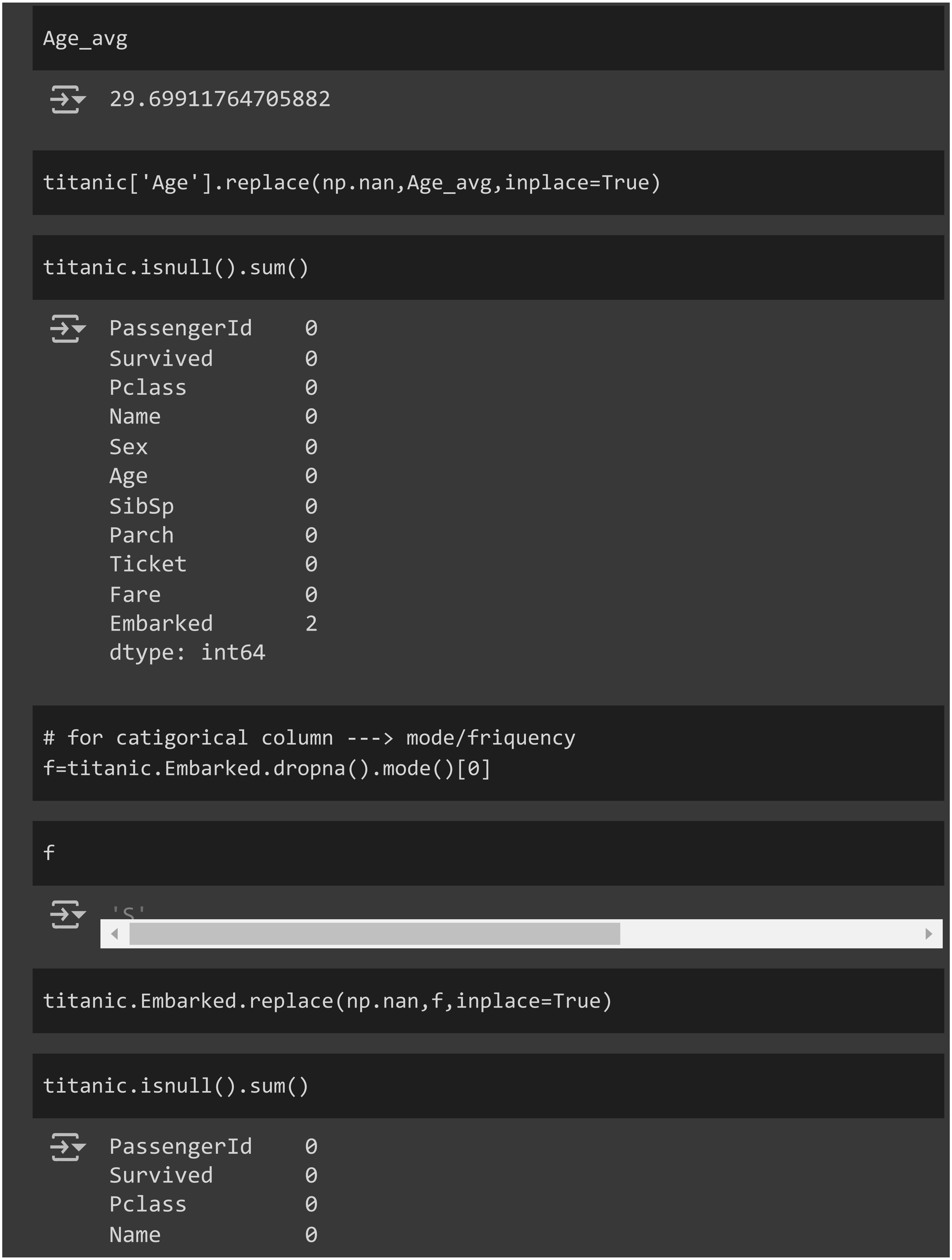


	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1

```
titanic.head(5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs	female	38.0	1

```
Age_avg=titanic.Age.mean()
```



```
Age_avg
```

```
29.69911764705882
```

```
titanic['Age'].replace(np.nan,Age_avg,inplace=True)
```

```
titanic.isnull().sum()
```

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

```
# for catigorical column ---> mode/friquency
f=titanic.Embarked.dropna().mode()[0]
```

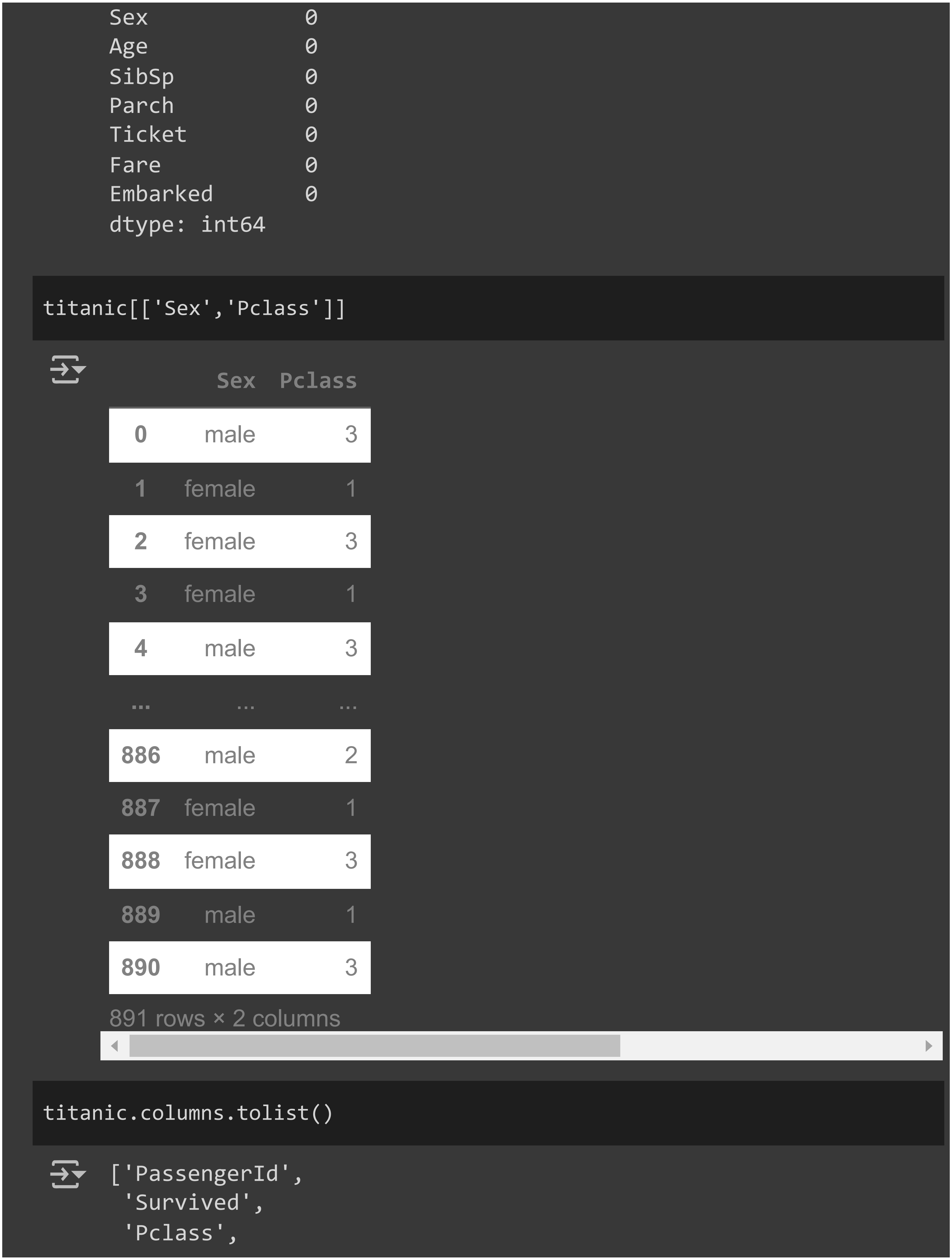
```
f
```

```
'S'
```

```
titanic.Embarked.replace(np.nan,f,inplace=True)
```

```
titanic.isnull().sum()
```

```
PassengerId    0
Survived       0
Pclass         0
Name           0
```



```
Sex             0
Age             0
SibSp           0
Parch           0
Ticket          0
Fare            0
Embarked        0
dtype: int64
```

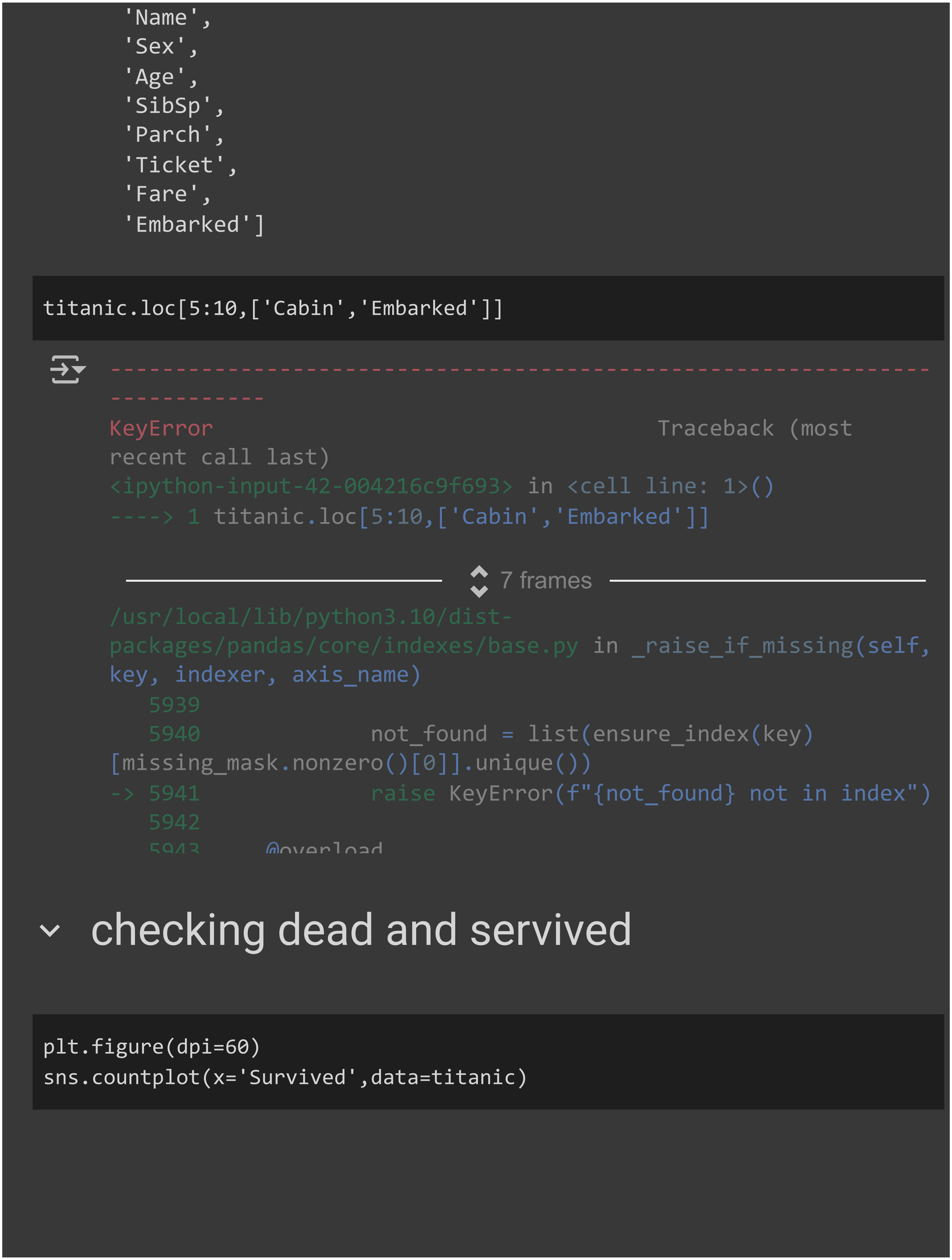
```
titanic[['Sex','Pclass']]
```

	Sex	Pclass
0	male	3
1	female	1
2	female	3
3	female	1
4	male	3
...	...	...
886	male	2
887	female	1
888	female	3
889	male	1
890	male	3

891 rows × 2 columns

```
titanic.columns.tolist()
```

```
['PassengerId',
 'Survived',
 'Pclass',
```



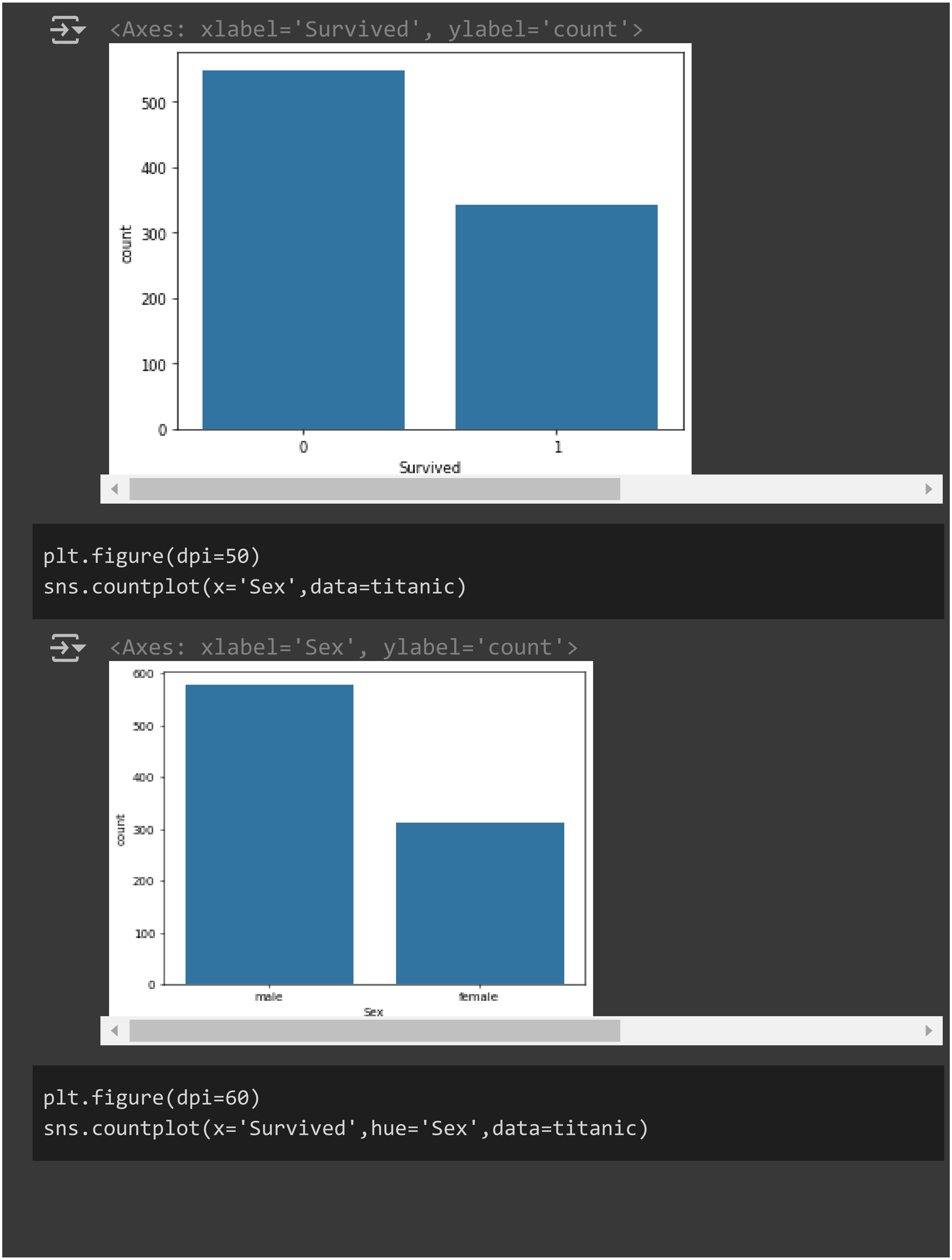
```
       'Name',
       'Sex',
       'Age',
       'SibSp',
       'Parch',
       'Ticket',
       'Fare',
       'Embarked']
```

```
titanic.loc[5:10,['Cabin','Embarked']]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-42-004216c9f693> in <cell line: 1>()
----> 1 titanic.loc[5:10,['Cabin','Embarked']]

7 frames

/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in _raise_if_missing(self, key, indexer, axis_name)
   5939 
   5940             not_found = list(ensure_index(key)[missing_mask.nonzero()[0]].unique())
-> 5941             raise KeyError(f"{not_found} not in index")
   5942 
   5943     @overload
```

checking dead and servived

```
plt.figure(dpi=60)
sns.countplot(x='Survived',data=titanic)
```



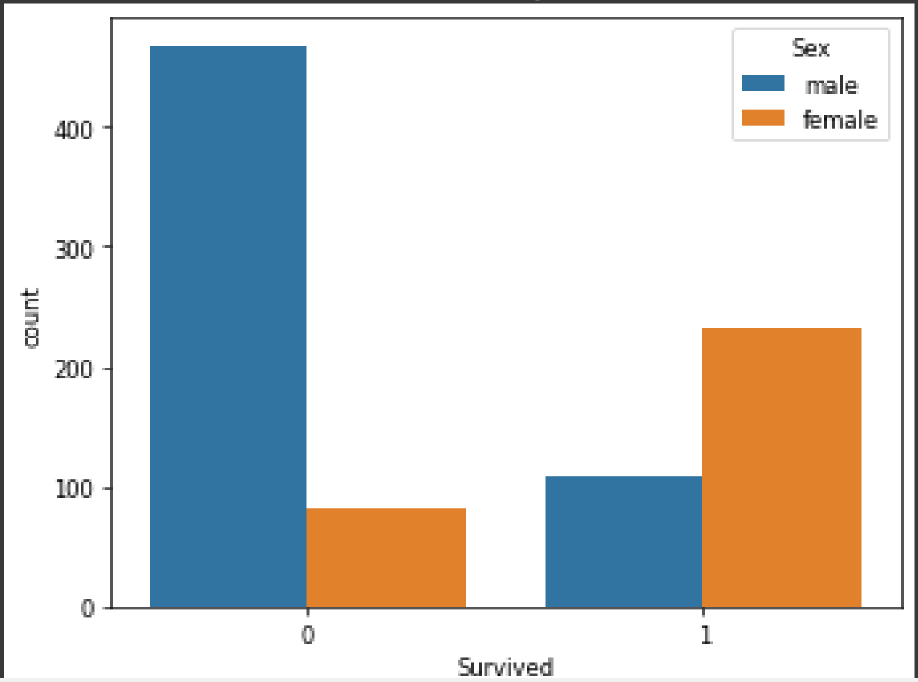
```
<Axes: xlabel='Survived', ylabel='count'>
```

```
plt.figure(dpi=50)
sns.countplot(x='Sex',data=titanic)
```

```
<Axes: xlabel='Sex', ylabel='count'>
```

```
plt.figure(dpi=60)
sns.countplot(x='Survived',hue='Sex',data=titanic)
```

```
<Axes: xlabel='Survived', ylabel='count'>
```



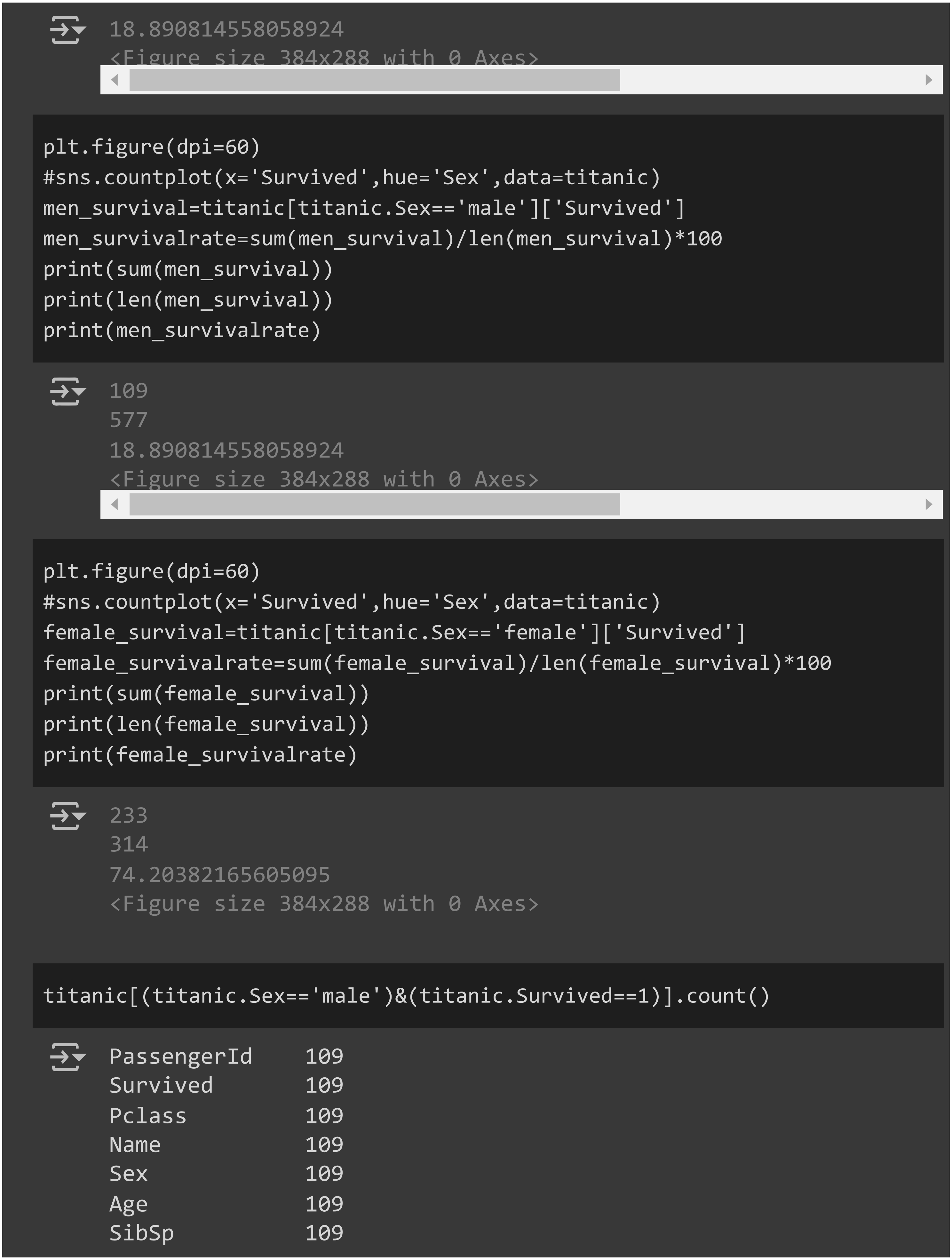
```
plt.figure(dpi=60)
#sns.countplot(x='Survived',hue='Sex',data=titanic)
men_survival=titanic[titanic.Sex=='male']['Survived'].count()
print(men_survival)
```

```
577
<Figure size 384x288 with 0 Axes>
```

filtering

survival rate for men

```
plt.figure(dpi=60)
#sns.countplot(x='Survived',hue='Sex',data=titanic)
men_survival=titanic[titanic.Sex=='male']['Survived']
men_survivalrate=sum(men_survival)/len(men_survival)*100
print(men_survivalrate)
```



```
18.890814558058924
<Figure size 384x288 with 0 Axes>
```

```
plt.figure(dpi=60)
#sns.countplot(x='Survived',hue='Sex',data=titanic)
men_survival=titanic[titanic.Sex=='male']['Survived']
men_survivalrate=sum(men_survival)/len(men_survival)*100
print(sum(men_survival))
print(len(men_survival))
print(men_survivalrate)
```

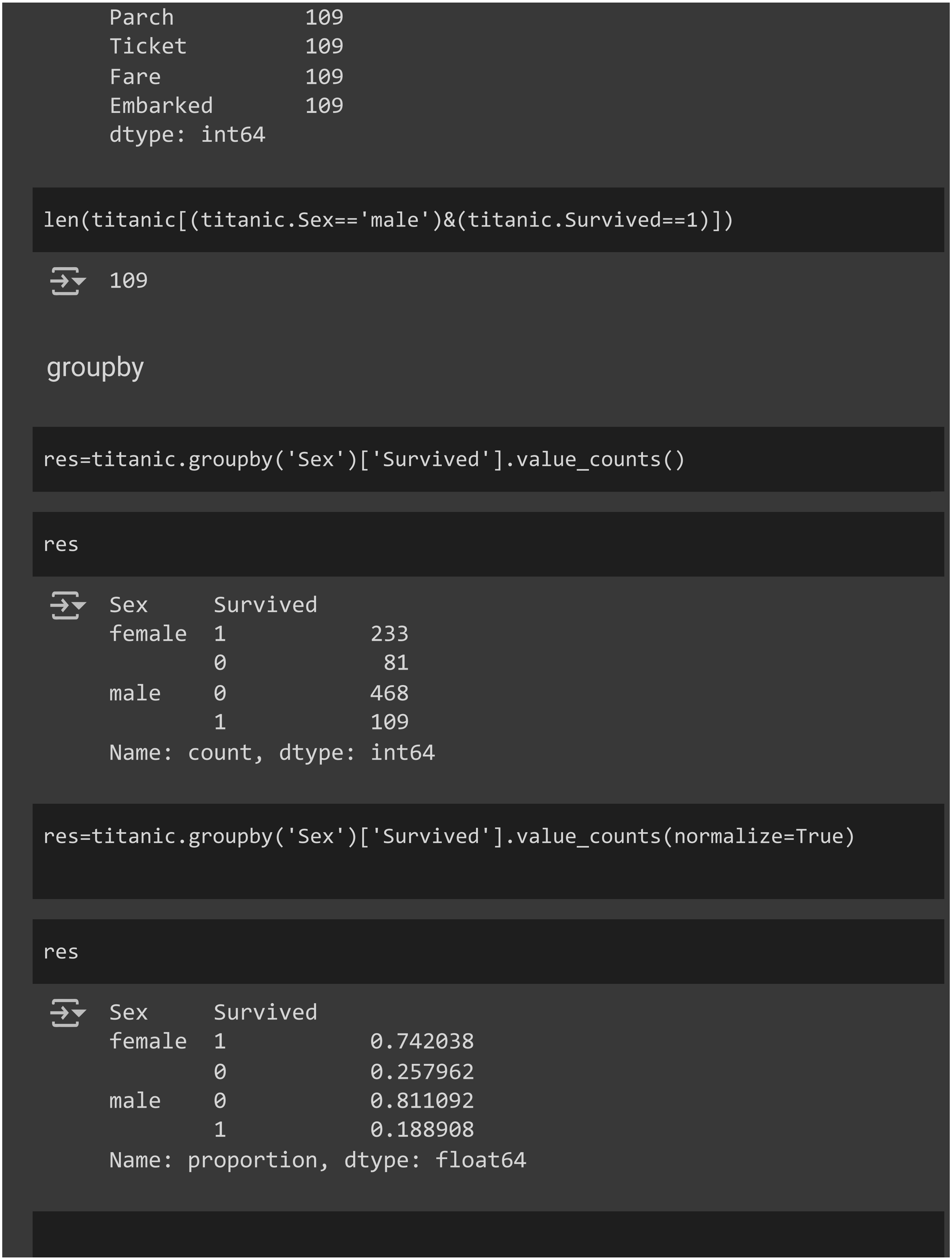
```
109
577
18.890814558058924
<Figure size 384x288 with 0 Axes>
```

```
plt.figure(dpi=60)
#sns.countplot(x='Survived',hue='Sex',data=titanic)
female_survival=titanic[titanic.Sex=='female']['Survived']
female_survivalrate=sum(female_survival)/len(female_survival)*100
print(sum(female_survival))
print(len(female_survival))
print(female_survivalrate)
```

```
233
314
74.20382165605095
<Figure size 384x288 with 0 Axes>
```

```
titanic[(titanic.Sex=='male')&(titanic.Survived==1)].count()
```

```
PassengerId    109
Survived       109
Pclass         109
Name           109
Sex            109
Age            109
SibSp          109
```



```
Parch          109
Ticket         109
Fare           109
Embarked       109
dtype: int64
```

```
len(titanic[(titanic.Sex=='male')&(titanic.Survived==1)])
```

```
109
```

groupby

```
res=titanic.groupby('Sex')['Survived'].value_counts()
```

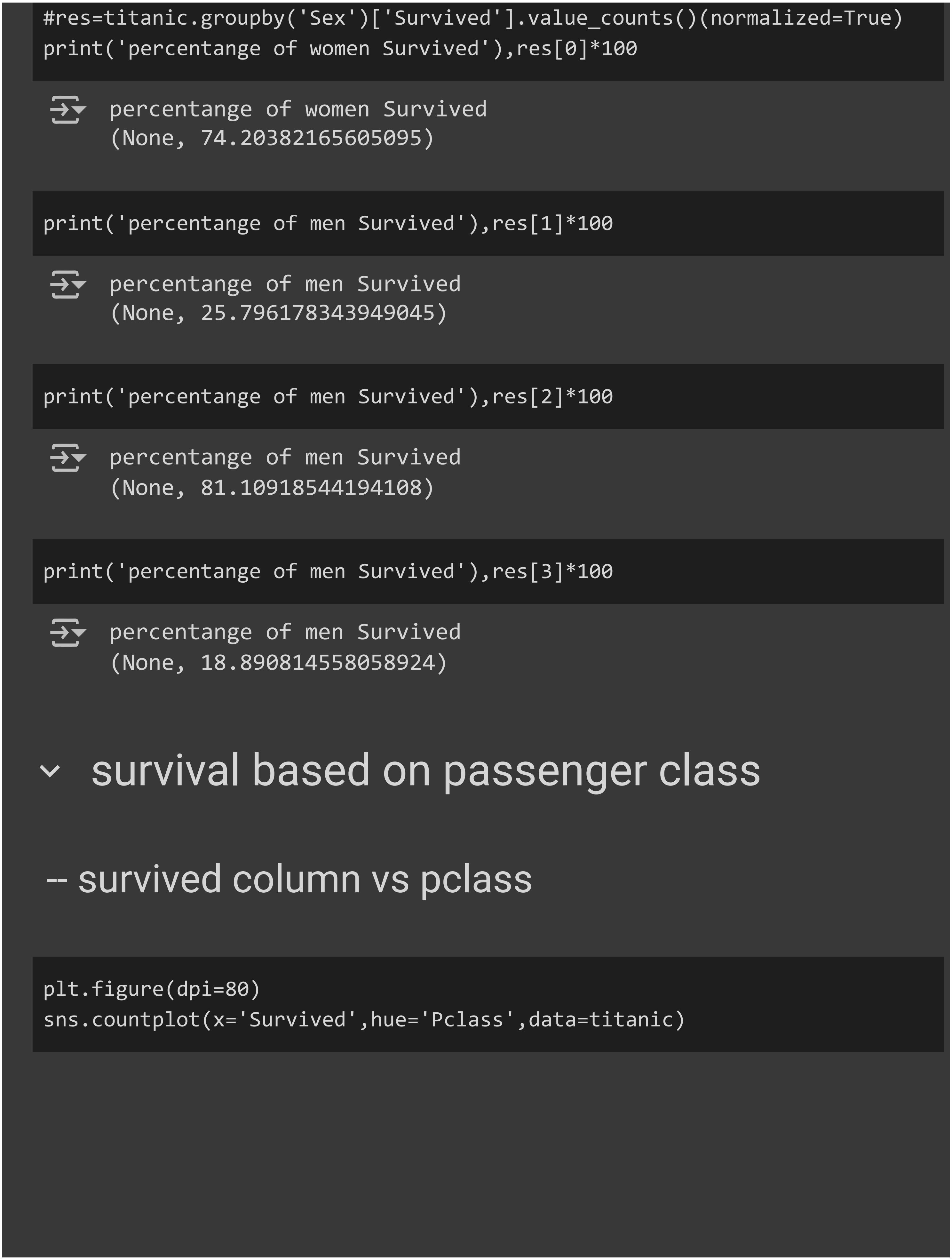
```
res
```

```
Sex     Survived
female  1           233
        0            81
male    0           468
        1           109
Name: count, dtype: int64
```

```
res=titanic.groupby('Sex')['Survived'].value_counts(normalize=True)
```

```
res
```

```
Sex     Survived
female  1           0.742038
        0           0.257962
male    0           0.811092
        1           0.188908
Name: proportion, dtype: float64
```



```
#res=titanic.groupby('Sex')['Survived'].value_counts()(normalized=True)
print('percentange of women Survived'),res[0]*100
```

```
percentange of women Survived
(None, 74.20382165605095)
```

```
print('percentange of men Survived'),res[1]*100
```

```
percentange of men Survived
(None, 25.796178343949045)
```

```
print('percentange of men Survived'),res[2]*100
```

```
percentange of men Survived
(None, 81.10918544194108)
```

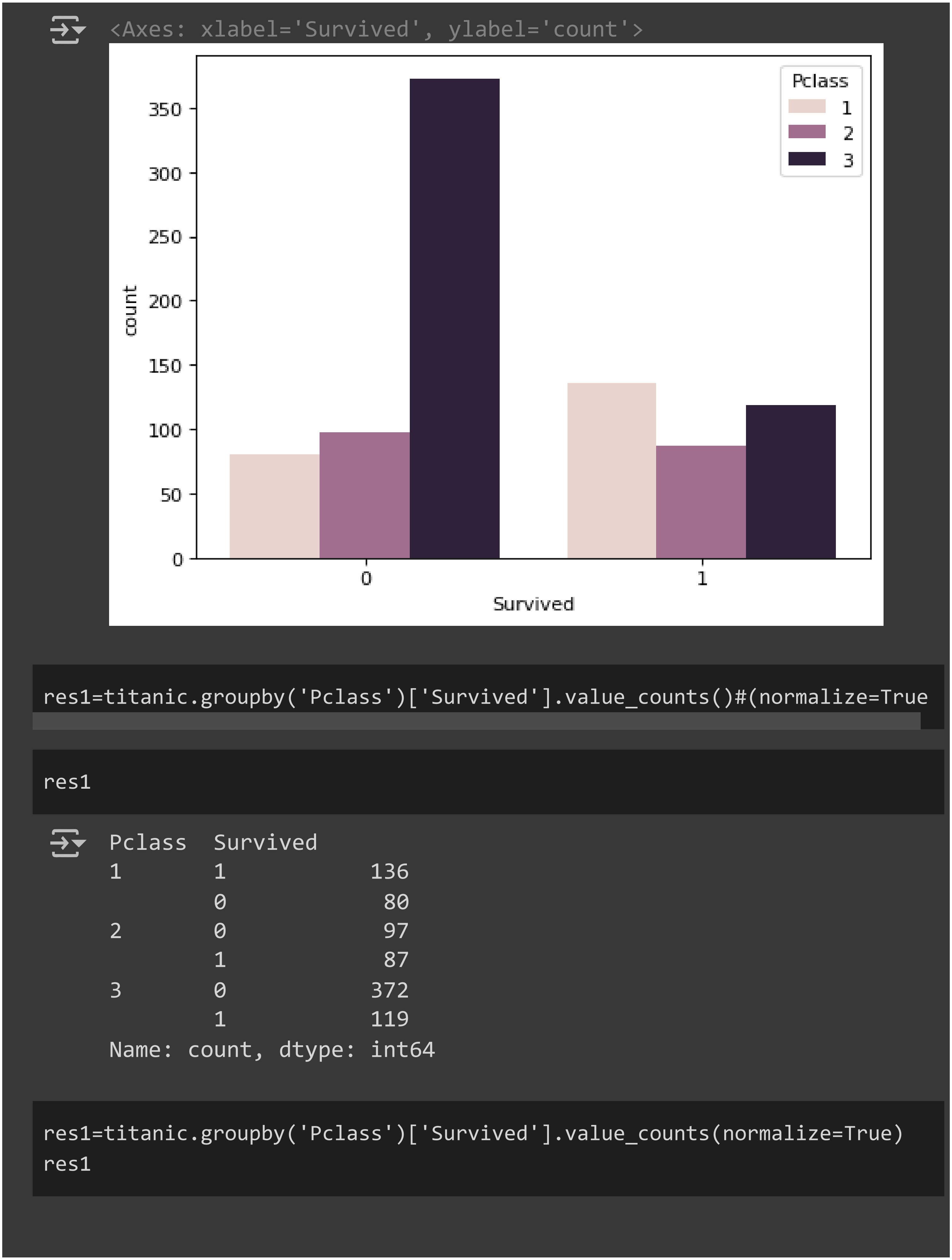
```
print('percentange of men Survived'),res[3]*100
```

```
percentange of men Survived
(None, 18.890814558058924)
```

survival based on passenger class

-- survived column vs pclass

```
plt.figure(dpi=80)
sns.countplot(x='Survived',hue='Pclass',data=titanic)
```



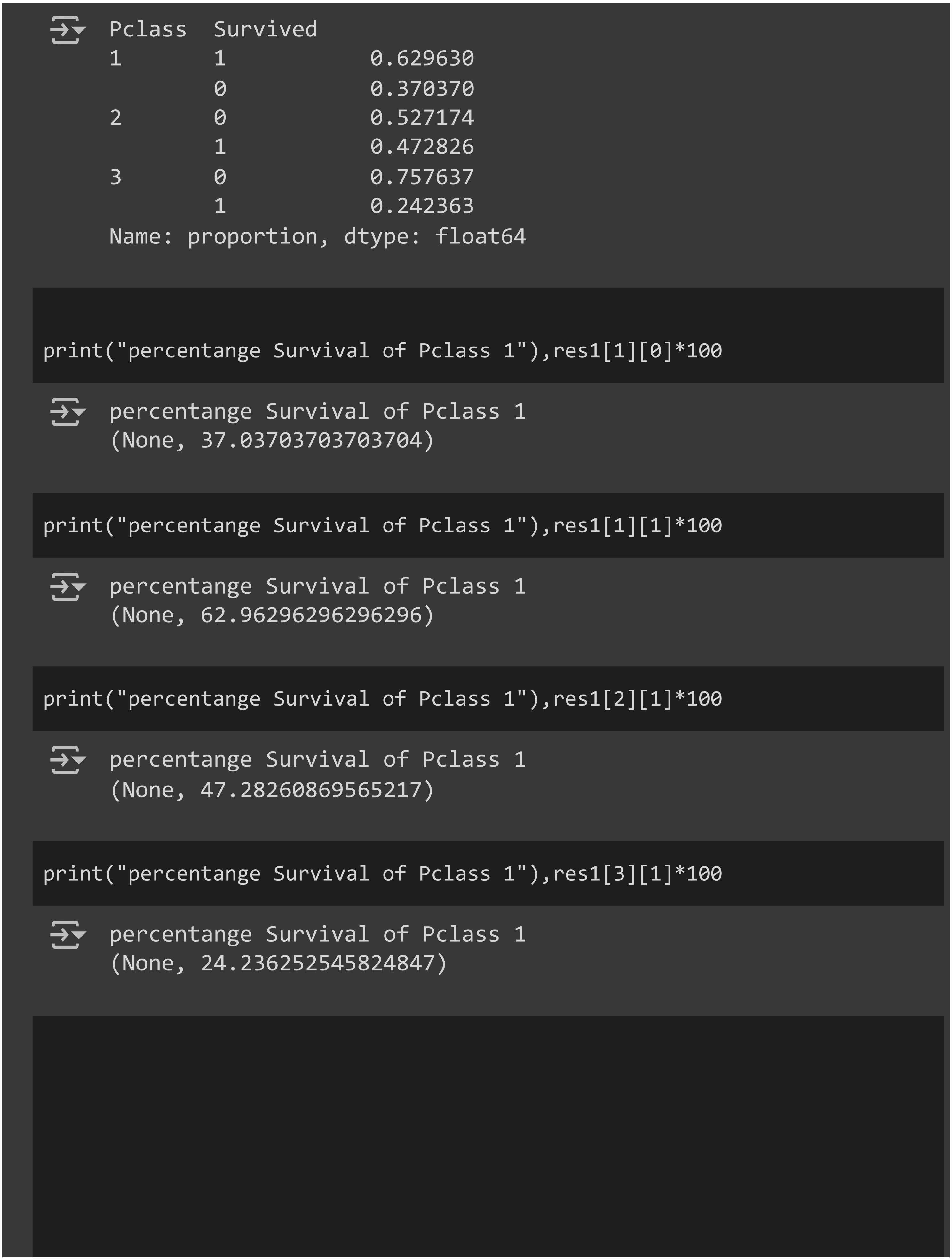
```
<Axes: xlabel='Survived', ylabel='count'>
```

```
res1=titanic.groupby('Pclass')['Survived'].value_counts()#(normalize=True
```

```
res1
```

```
Pclass  Survived
1       1           136
        0            80
2       0            97
        1            87
3       0           372
        1           119
Name: count, dtype: int64
```

```
res1=titanic.groupby('Pclass')['Survived'].value_counts(normalize=True)
res1
```



```
Pclass  Survived
1       1           0.629630
        0           0.370370
2       0           0.527174
        1           0.472826
3       0           0.757637
        1           0.242363
Name: proportion, dtype: float64
```

```
print("percentange Survival of Pclass 1"),res1[1][0]*100
```

```
percentange Survival of Pclass 1
(None, 37.03703703703704)
```

```
print("percentange Survival of Pclass 1"),res1[1][1]*100
```

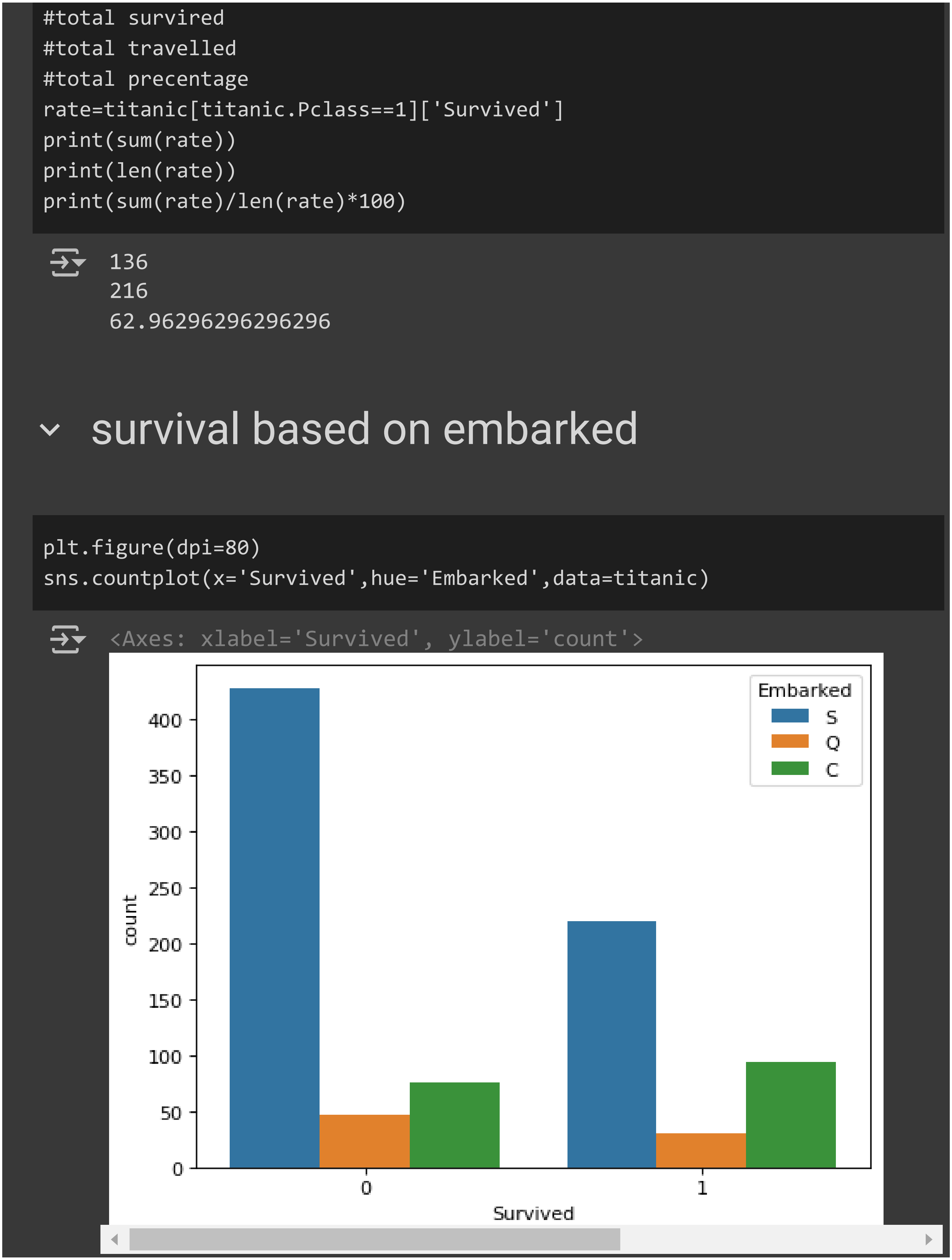
```
percentange Survival of Pclass 1
(None, 62.96296296296296)
```

```
print("percentange Survival of Pclass 1"),res1[2][1]*100
```

```
percentange Survival of Pclass 1
(None, 47.28260869565217)
```

```
print("percentange Survival of Pclass 1"),res1[3][1]*100
```

```
percentange Survival of Pclass 1
(None, 24.236252545824847)
```



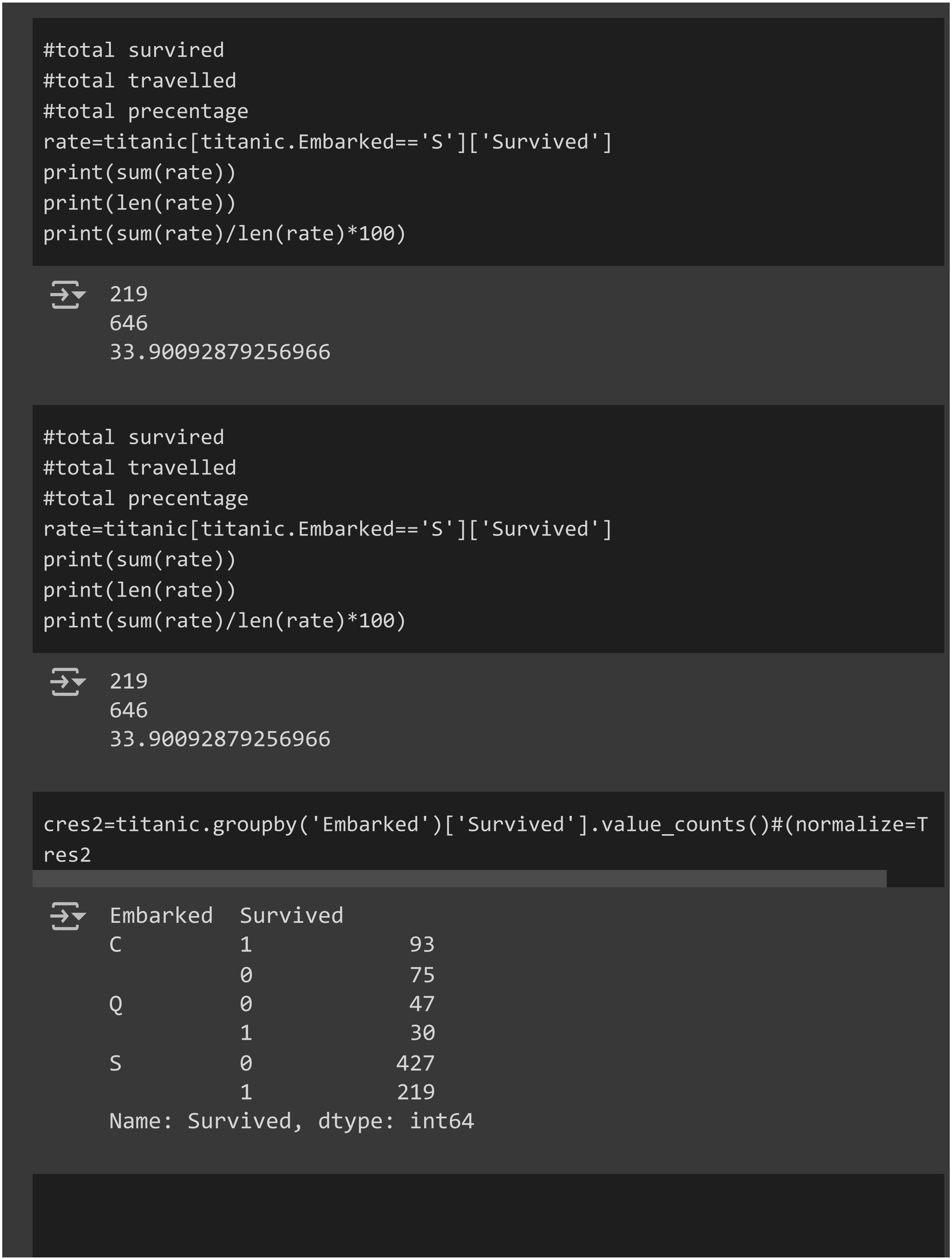
```
#total survired
#total travelled
#total precentage
rate=titanic[titanic.Pclass==1]['Survived']
print(sum(rate))
print(len(rate))
print(sum(rate)/len(rate)*100)
```

```
136
216
62.96296296296296
```

survival based on embarked

```
plt.figure(dpi=80)
sns.countplot(x='Survived',hue='Embarked',data=titanic)
```

```
<Axes: xlabel='Survived', ylabel='count'>
```



```
#total survired
#total travelled
#total precentage
rate=titanic[titanic.Embarked=='S']['Survived']
print(sum(rate))
print(len(rate))
print(sum(rate)/len(rate)*100)
```

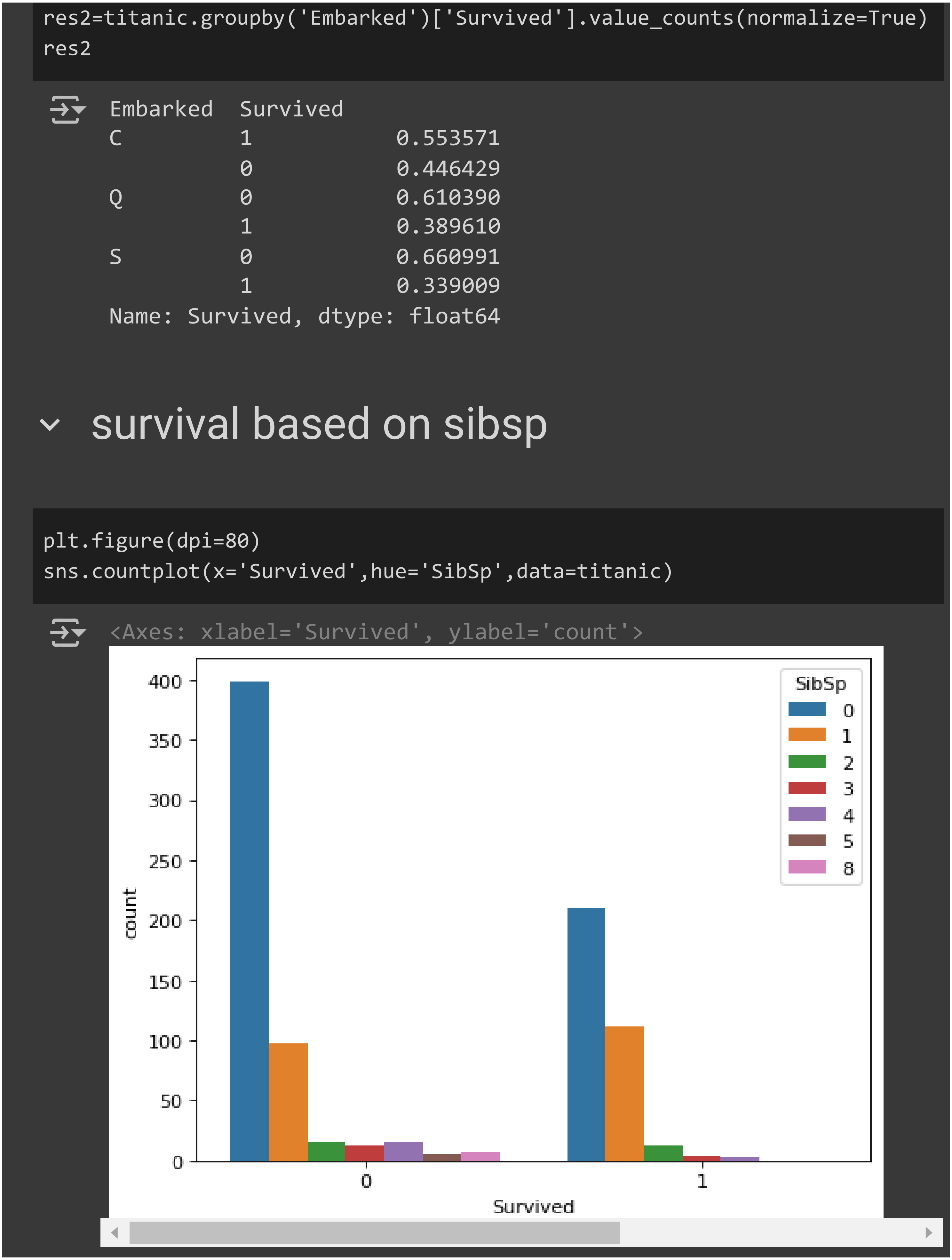
```
219
646
33.90092879256966
```

```
#total survired
#total travelled
#total precentage
rate=titanic[titanic.Embarked=='S']['Survived']
print(sum(rate))
print(len(rate))
print(sum(rate)/len(rate)*100)
```

```
219
646
33.90092879256966
```

```
cres2=titanic.groupby('Embarked')['Survived'].value_counts()#(normalize=T
res2
```

```
Embarked  Survived
C         1            93
          0            75
Q         0            47
          1            30
S         0           427
          1           219
Name: Survived, dtype: int64
```



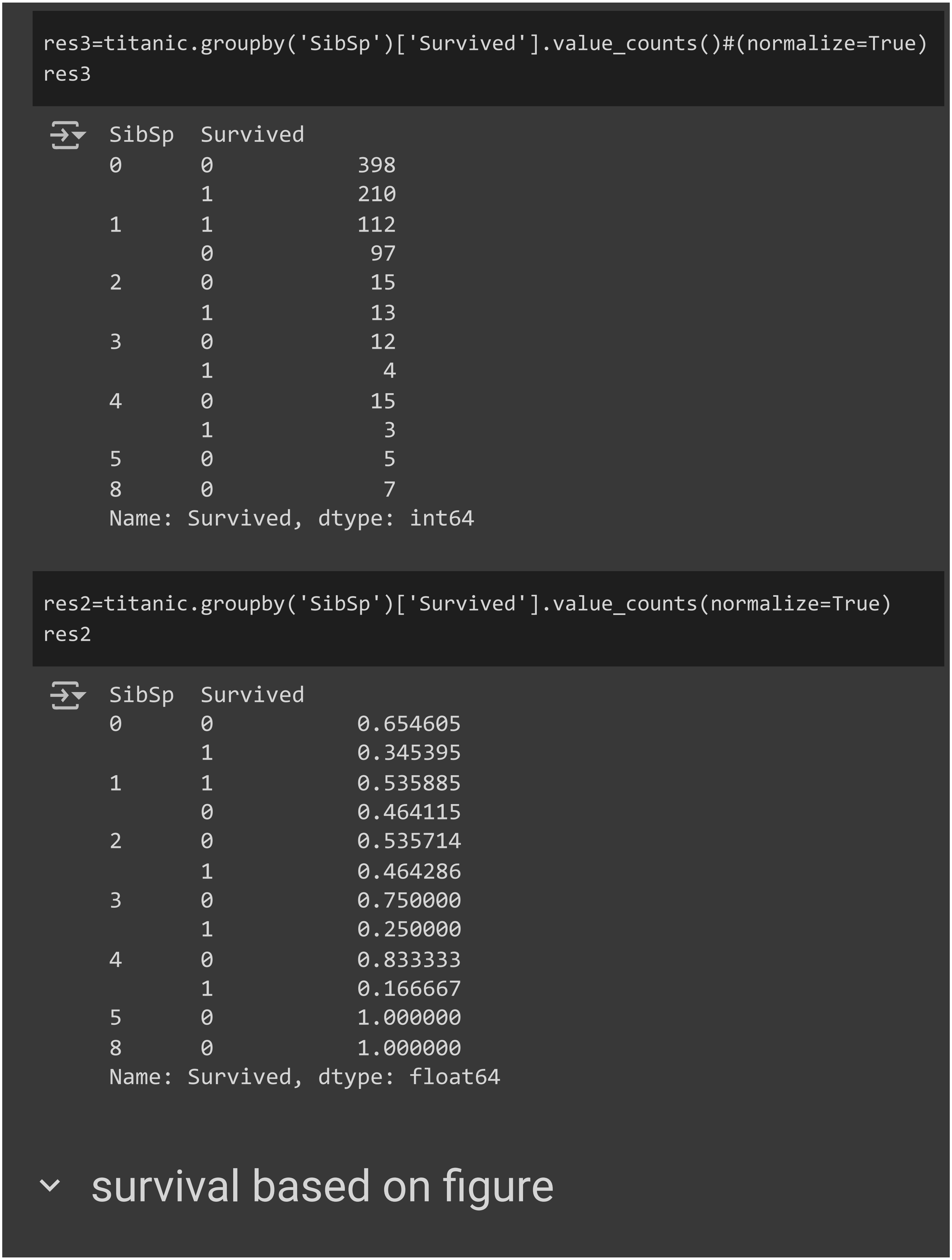
```
res2=titanic.groupby('Embarked')['Survived'].value_counts(normalize=True)
res2
```

```
Embarked  Survived
C         1           0.553571
          0           0.446429
Q         0           0.610390
          1           0.389610
S         0           0.660991
          1           0.339009
Name: Survived, dtype: float64
```

survival based on sibsp

```
plt.figure(dpi=80)
sns.countplot(x='Survived',hue='SibSp',data=titanic)
```

```
<Axes: xlabel='Survived', ylabel='count'>
```



```
res3=titanic.groupby('SibSp')['Survived'].value_counts()#(normalize=True)
res3
```

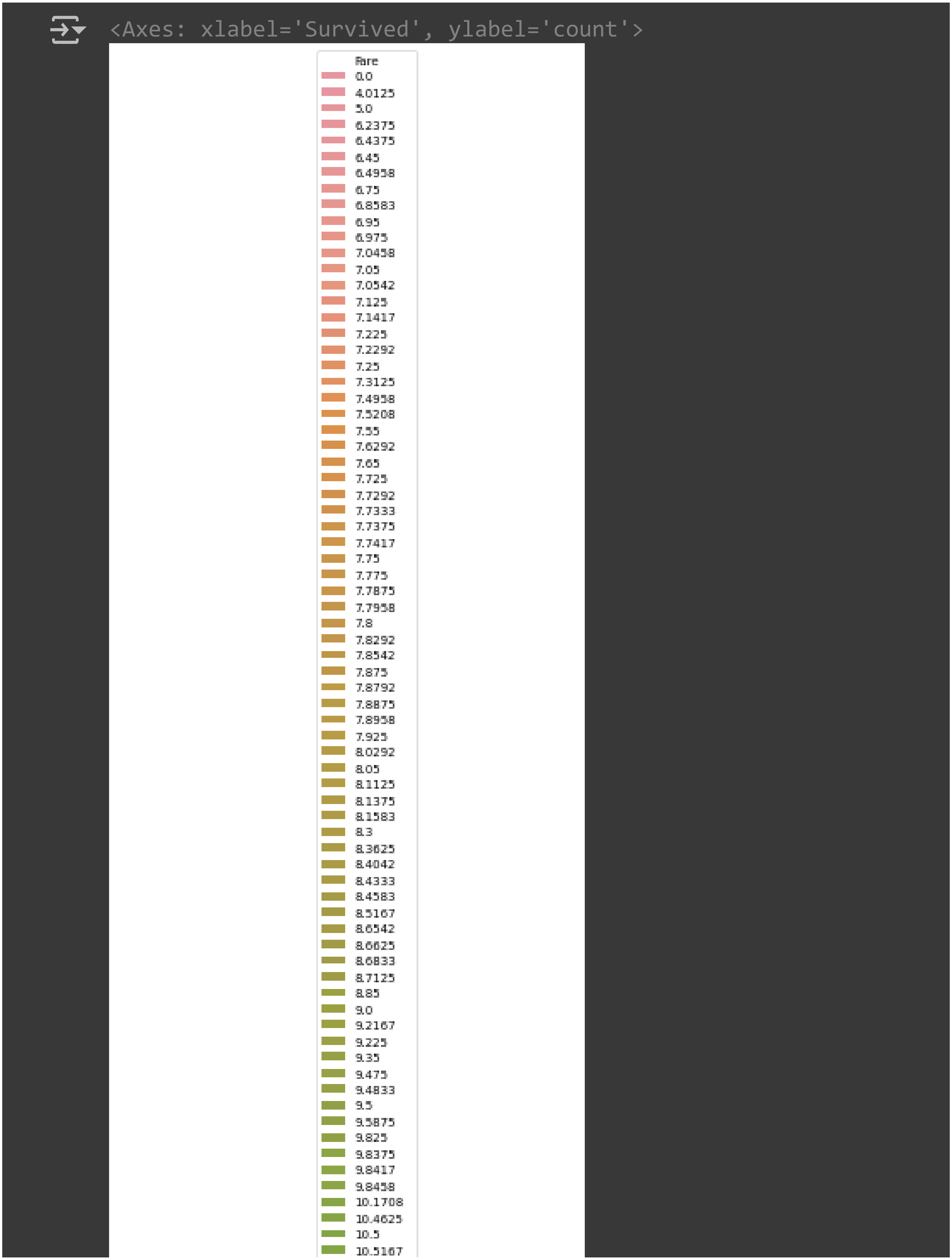
```
SibSp  Survived
0      0           398
       1           210
1      1           112
       0            97
2      0            15
       1            13
3      0            12
       1             4
4      0            15
       1             3
5      0             5
8      0             7
Name: Survived, dtype: int64
```

```
res2=titanic.groupby('SibSp')['Survived'].value_counts(normalize=True)
res2
```

```
SibSp  Survived
0      0           0.654605
       1           0.345395
1      1           0.535885
       0           0.464115
2      0           0.535714
       1           0.464286
3      0           0.750000
       1           0.250000
4      0           0.833333
       1           0.166667
5      0           1.000000
8      0           1.000000
Name: Survived, dtype: float64
```

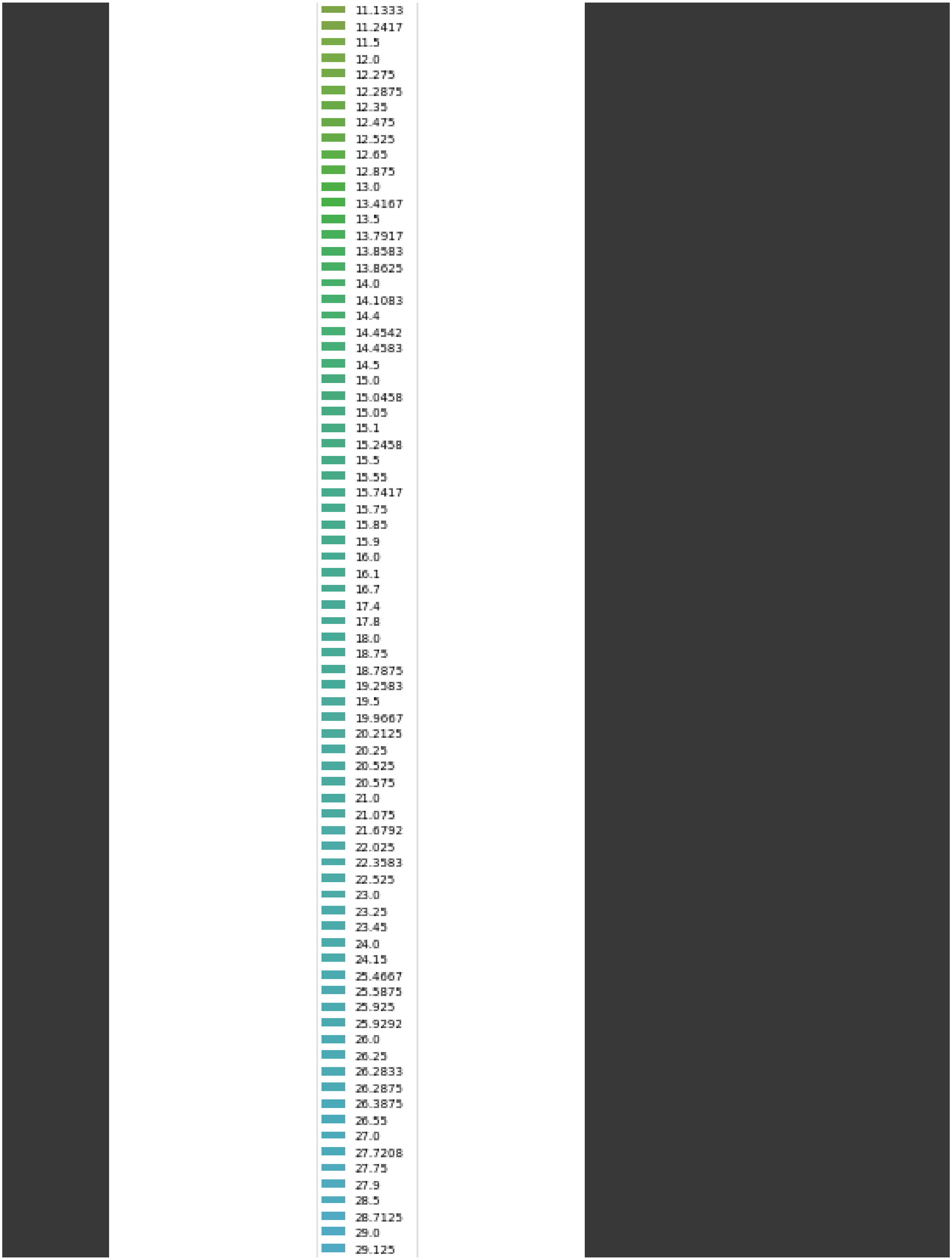
survival based on figure

```
plt.figure(dpi=50)
sns.countplot(x='Survived',hue='Fare',data=titanic)
```

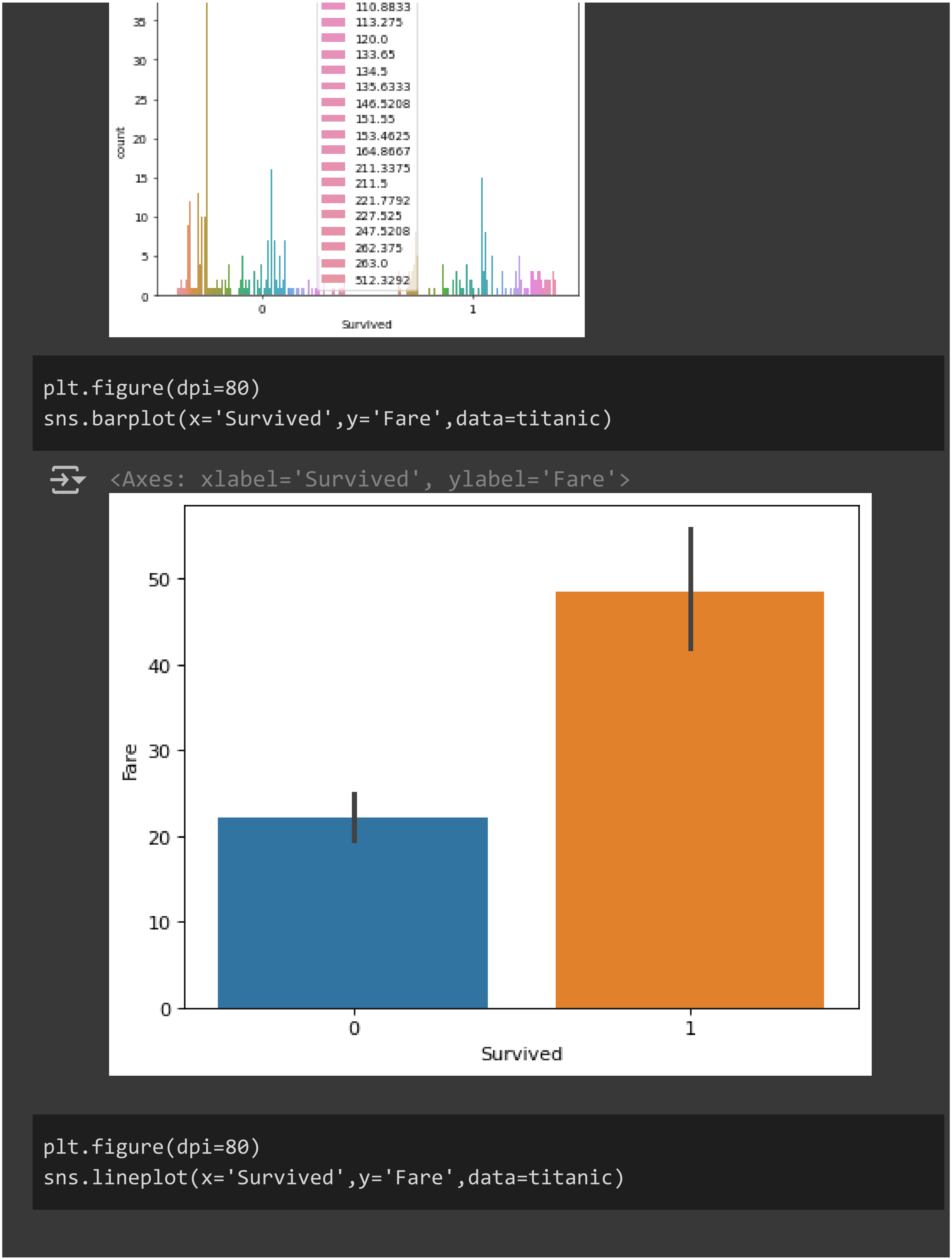


```
<Axes: xlabel='Survived', ylabel='count'>
```

Fare
0.0
4.0125
5.0
6.2375
6.4375
6.45
6.4958
6.75
6.8583
6.95
6.975
7.0458
7.05
7.0542
7.125
7.1417
7.225
7.2292
7.25
7.3125
7.4958
7.5208
7.55
7.6292
7.65
7.725
7.7292
7.7333
7.7375
7.7417
7.75
7.775
7.7875
7.7958
7.8
7.8292
7.8542
7.875
7.8792
7.8875
7.8958
7.925
8.0292
8.05
8.1125
8.1375
8.1583
8.3
8.3625
8.4042
8.4333
8.4583
8.5167
8.6542
8.6625
8.6833
8.7125
8.85
9.0
9.2167
9.225
9.35
9.475
9.4833
9.5
9.5875
9.825
9.8375
9.8417
9.8458
10.1708
10.4625
10.5
10.5167


11.1333
11.2417
11.5
12.0
12.275
12.2875
12.35
12.475
12.525
12.65
12.875
13.0
13.4167
13.5
13.7917
13.8583
13.8625
14.0
14.1083
14.4
14.4542
14.4583
14.5
15.0
15.0458
15.05
15.1
15.2458
15.5
15.55
15.7417
15.75
15.85
15.9
16.0
16.1
16.7
17.4
17.8
18.0
18.75
18.7875
19.2583
19.5
19.9667
20.2125
20.25
20.525
20.575
21.0
21.075
21.6792
22.025
22.3583
22.525
23.0
23.25
23.45
24.0
24.15
25.4667
25.5875
25.925
25.9292
26.0
26.25
26.2833
26.2875
26.3875
26.55
27.0
27.7208
27.75
27.9
28.5
28.7125
29.0
29.125

29.7
30.0
30.0708
30.5
30.6958
31.0
31.275
31.3875
32.3208
32.5
33.0
33.5
34.0208
34.375
34.6542
35.0
35.5
36.75
37.0042
38.5
39.0
39.4
39.6
39.6875
40.125
41.5792
42.4
46.9
47.1
49.5
49.5042
50.0
50.4958
51.4792
51.8625
52.0
52.5542
53.1
55.0
55.4417
55.9
56.4958
56.9292
57.0
57.9792
59.4
61.175
61.3792
61.9792
63.3583
65.0
66.6
69.3
69.55
71.0
71.2833
73.5
75.25
76.2917
76.7292
77.2875
77.9583
78.2667
78.85
79.2
79.65
80.0
81.8583
82.1708
83.1583
83.475
86.5
89.1042
90.0
91.0792
93.5
106.425
108.9

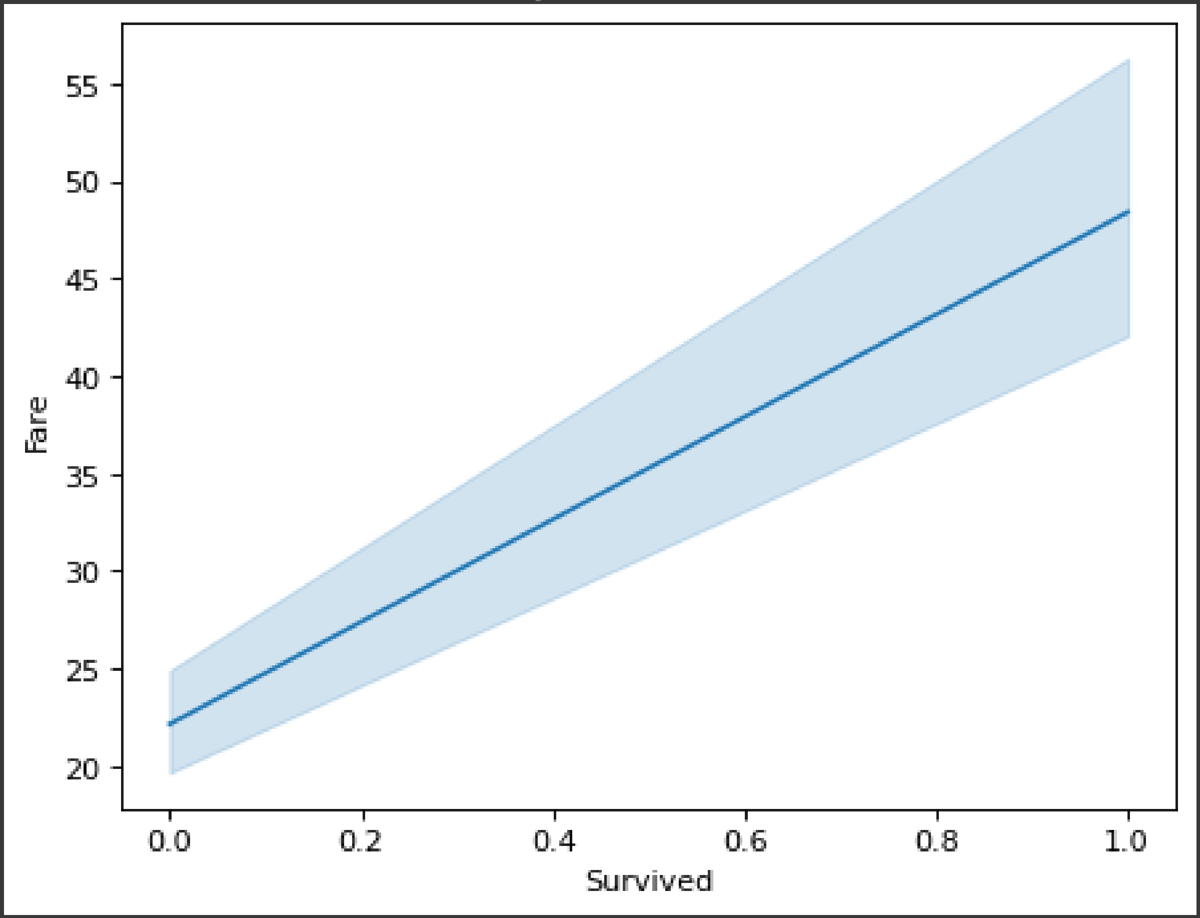


```
plt.figure(dpi=80)
sns.barplot(x='Survived',y='Fare',data=titanic)
```

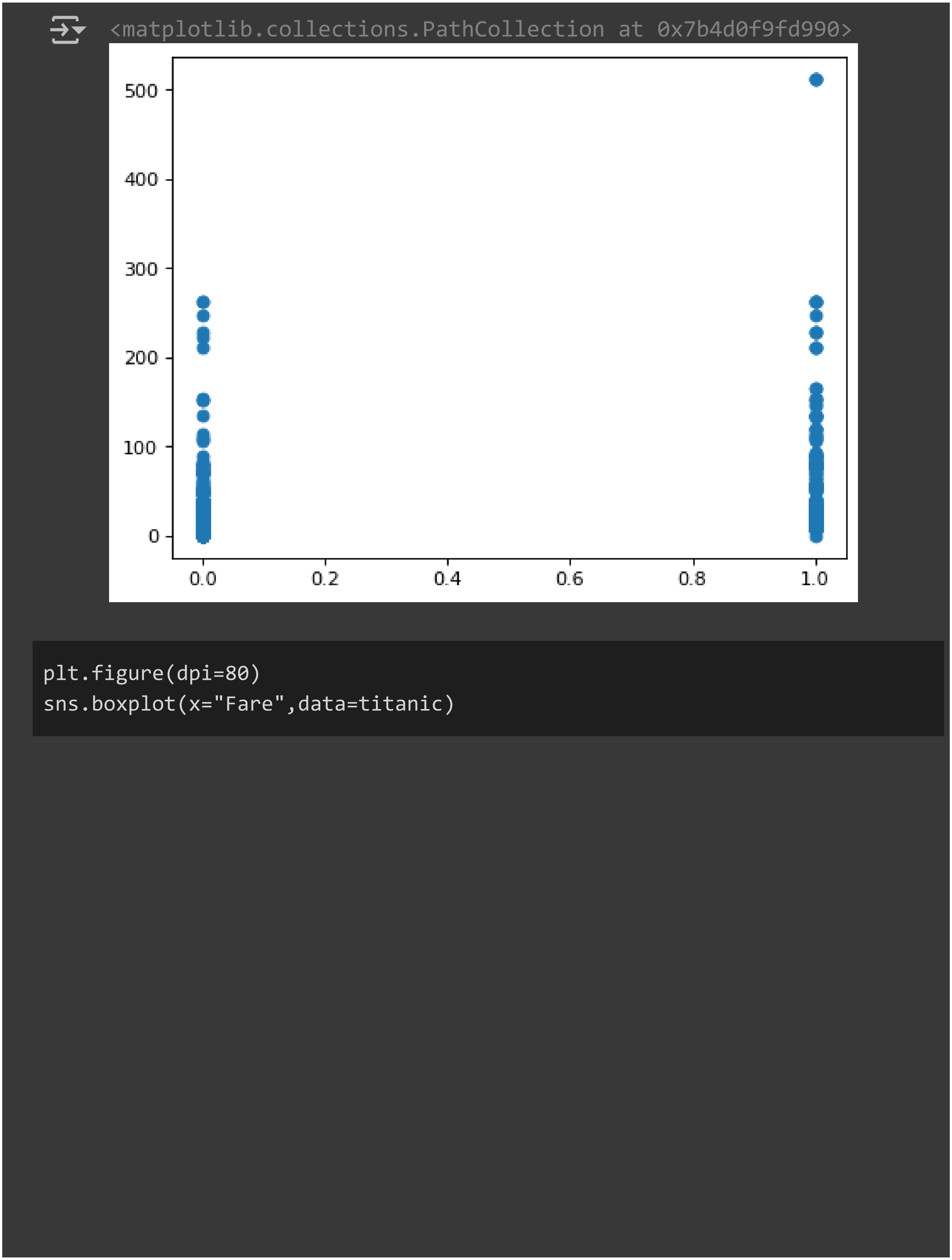
```
<Axes: xlabel='Survived', ylabel='Fare'>
```

```
plt.figure(dpi=80)
sns.lineplot(x='Survived',y='Fare',data=titanic)
```

```
<Axes: xlabel='Survived', ylabel='Fare'>
```



```
x=titanic["Survived"]
y=titanic["Fare"]
plt.figure(dpi=80)
plt.scatter(x,y)
```



```
<matplotlib.collections.PathCollection at 0x7b4d0f9fd990>
```

```
plt.figure(dpi=80)
sns.boxplot(x="Fare",data=titanic)
```

```
<Axes: xlabel='Fare'>
```

survived vs age

```
plt.figure(dpi=80)
```