

```
In [2]: import seaborn as sns
import matplotlib.pyplot as plt

In [3]: titanic=sns.load_dataset('titanic')
titanic

Out[3]:
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	NaN	Southampton	no	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	B	Southampton	yes	True
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	NaN	Southampton	no	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	C	Cherbourg	yes	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	NaN	Queenstown	no	True

891 rows × 15 columns

```
In [7]: #1. data understanding
titanic.head()

Out[7]:
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

```
In [8]: titanic.describe()

Out[8]:
```

	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
In [5]: #2. Initial check up
titanic.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   survived              891 non-null    int64
 1   pclass                891 non-null    int64
 2   sex                  891 non-null    object
 3   age                  714 non-null    float64
 4   sibsp                891 non-null    int64
 5   parch               891 non-null    int64
 6   fare                 891 non-null    float64
 7   embarked             889 non-null    object
 8   class                891 non-null    category
 9   who                  891 non-null    object
10  adult_male           891 non-null    bool
11  deck                203 non-null    category
12  embark_town         889 non-null    object
13  alive               891 non-null    object
14  alone              891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB

In [ ]: #asking Questions
#1. age wise survival
#2. gender wise survival
#3. passenger wise survival
#4. fare wise survival
#5. class wise sex
#6. fare wise alone
#7. age wise alive
#8. alone wise class
#9. age wise alive
#10. fare wise age
```

```
In [10]: titanic['age'].value_counts(normalize=True)*100

Out[10]:
```

age	count	proportion
24.00	4.291681	0.478688
22.00	3.781513	0.424567
18.00	3.641457	0.404632
19.00	3.591401	0.399140
28.00	3.561401	0.394140
...	...	...
36.50	0.140956	0.156523
55.50	0.140956	0.156523
9.92	0.140956	0.156523
23.50	0.140956	0.156523
74.00	0.140956	0.156523

Name: proportion, Length: 88, dtype: float64

```
In [11]: titanic['sex'].value_counts(normalize=True)*100

Out[11]:
```

sex	count	proportion
male	64.758698	0.726253
female	55.241302	0.617747

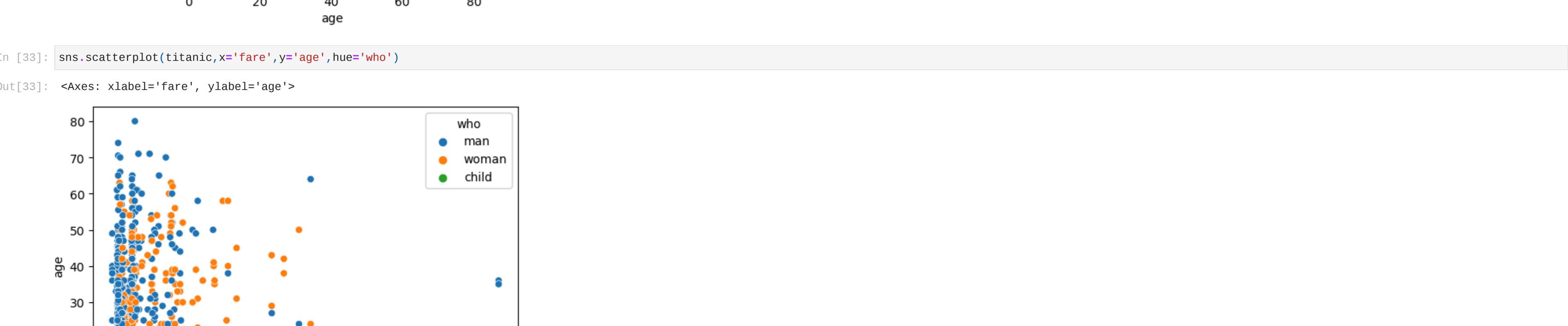
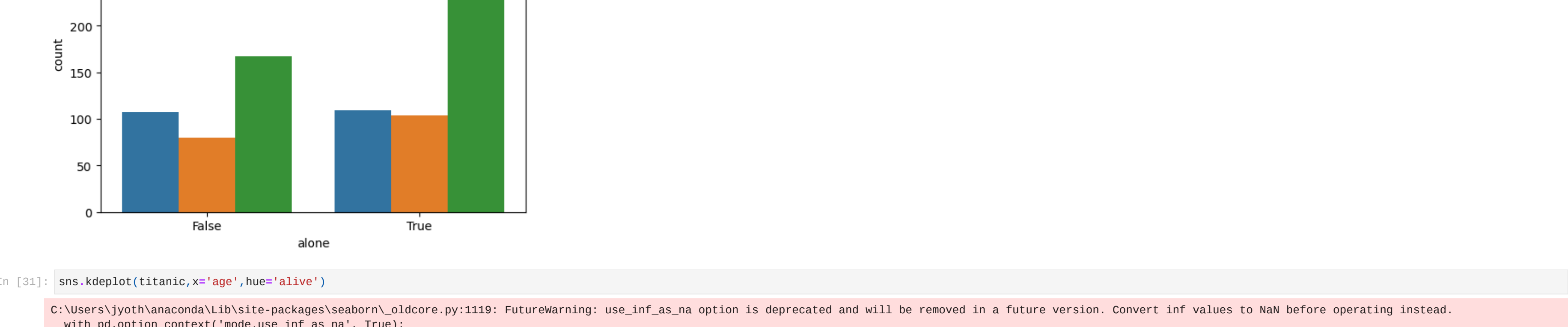
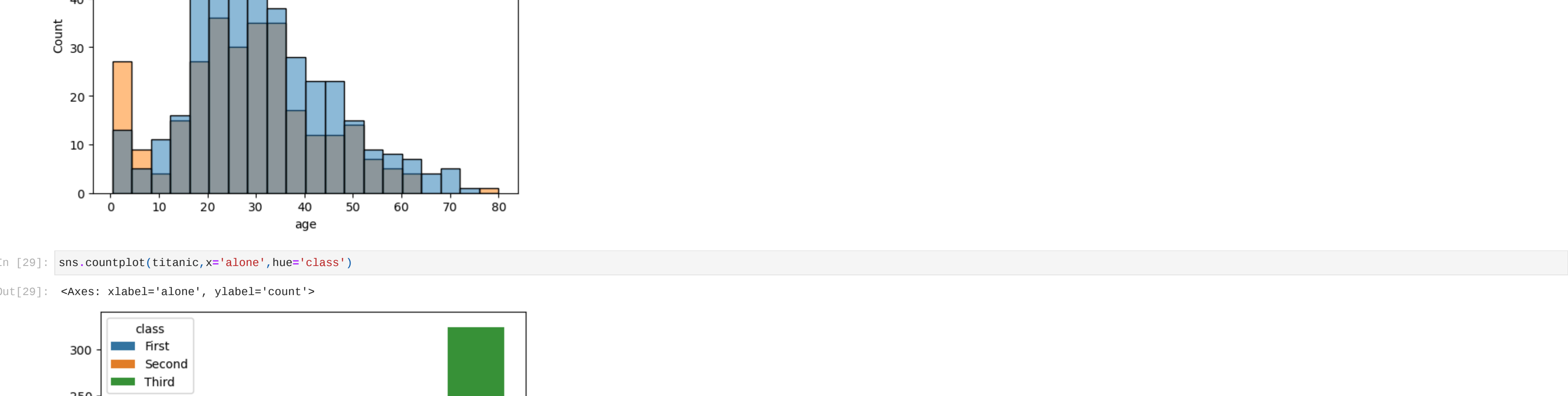
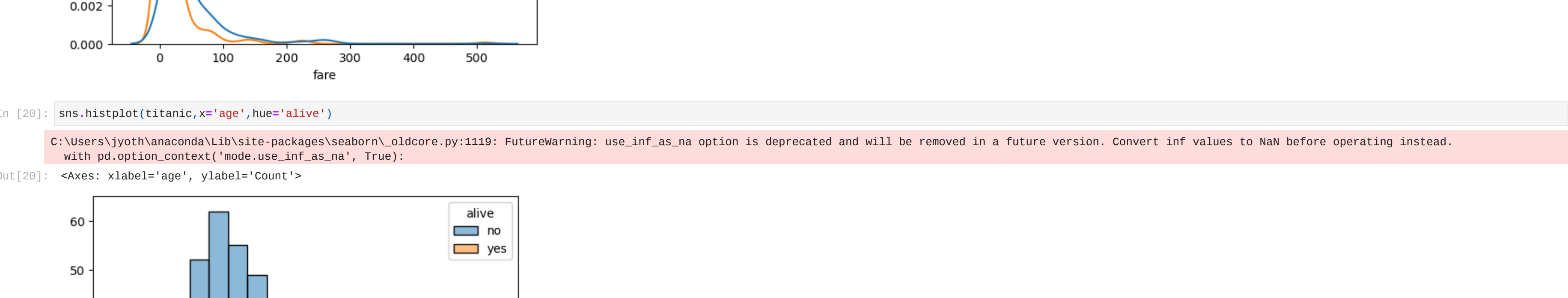
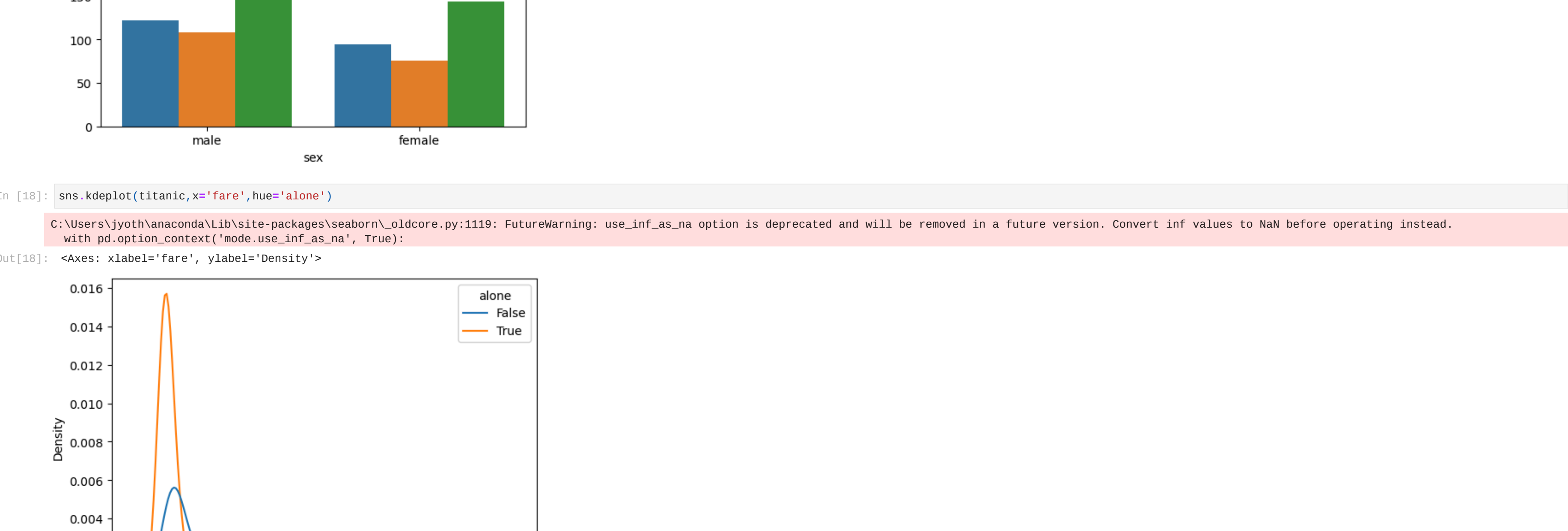
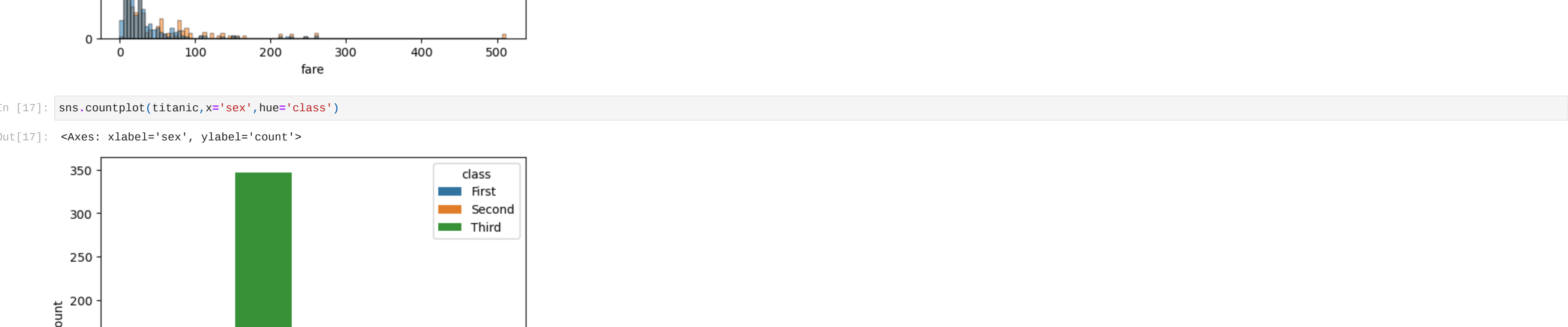
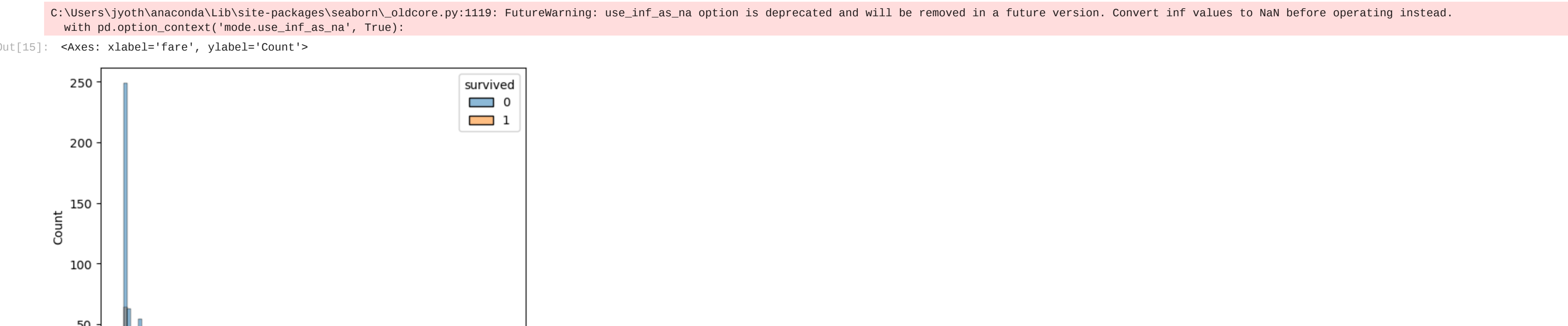
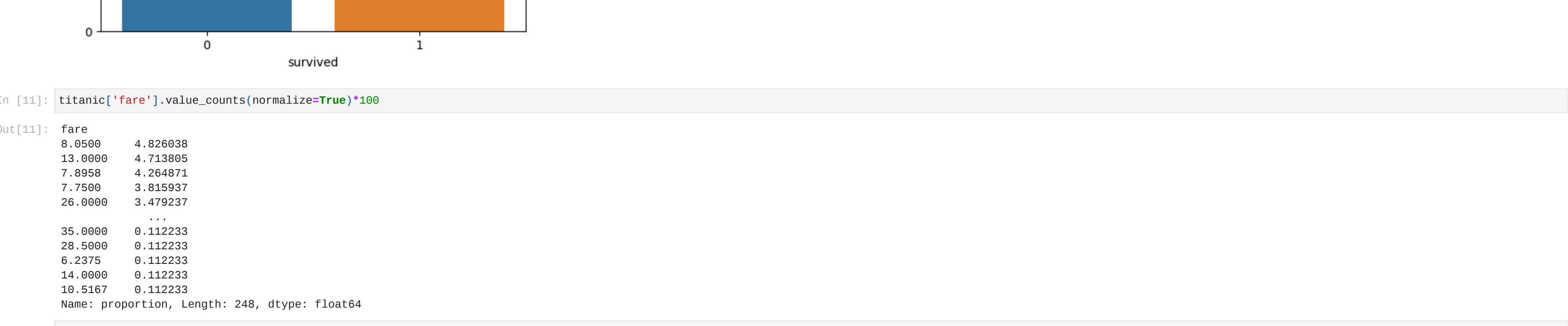
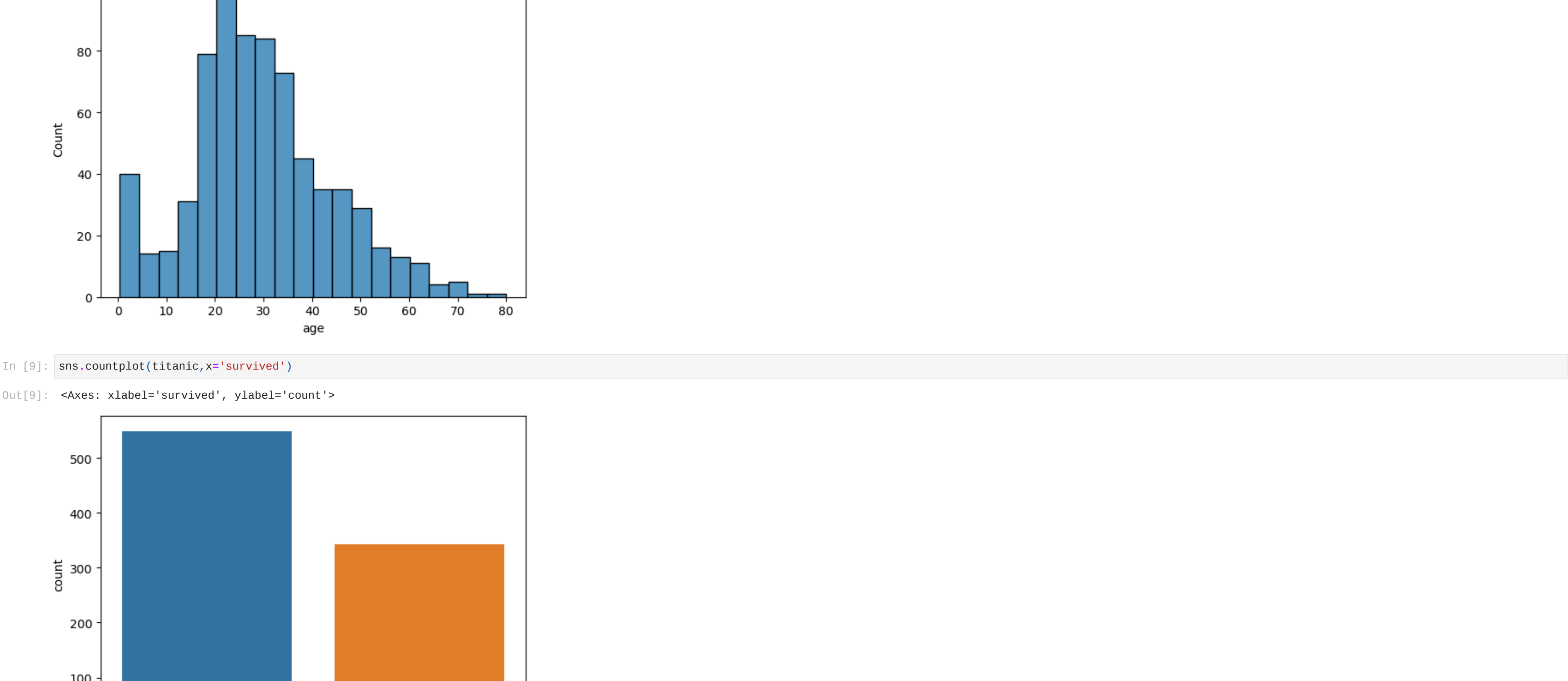
Name: proportion, dtype: float64

```
In [12]: titanic['survived'].value_counts(normalize=True)*100

Out[12]:
```

survived	count	proportion
0	61.616162	0.691471
1	38.383838	0.428529

Name: proportion, dtype: float64



In [ ]: #insights

#we have to import the dataset and perform the head(),describe(),info() on that dataset

#24 age people survived of 4% and 36 age people survived of 0.1% and the least people are survived at 74 age

#out of 100% the 64% male are survived and 35% female are survived

#500 above people are survived and 300 people are not survived

#highest fare is 10 and count is 250 the lowest fare is between 100 to 200

#in male the highest count is in third class at 350 in female the highest count is in third class at 150

#above 60 people are not alive at the age 25, the 28 people are alive at the age 3

#the highest people are alone in third and class and the lowest people are not alone in second class

#out of 100% the 55% people are alive at the age 36 and 45% people are not alive at the age 22

#the highest fare is men at the age 80 and the lowest fare is child at the age 0