```
import seaborn as sns
```

```
sns.get_dataset_names()
```

```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

```
df=sns.load_dataset('titanic')
```

```
df
```

Show hidden output

Next steps: Generate code with df New interactive sheet

```
df=sns.load_dataset('tips')
```

```
df
```

```
     total_bill   tip     sex  smoker   day    time  size
0         16.99  1.01  Female      No   Sun  Dinner     2
1         10.34  1.66    Male      No   Sun  Dinner     3
2         21.01  3.50    Male      No   Sun  Dinner     3
3         23.68  3.31    Male      No   Sun  Dinner     2
4         24.59  3.61  Female      No   Sun  Dinner     4
...         ...   ...     ...     ...   ...     ...   ...
239       29.03  5.92    Male      No   Sat  Dinner     3
240       27.18  2.00  Female     Yes   Sat  Dinner     2
241       22.67  2.00    Male     Yes   Sat  Dinner     2
242       17.82  1.75    Male      No   Sat  Dinner     2
243       18.78  3.00  Female      No  Thur  Dinner     2
244 rows × 7 columns
```

Next steps: Generate code with df New interactive sheet

```
df=sns.load_dataset('titanic')
```

```
df
```

	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

Next steps: Generate code with df New interactive sheet

```
import pandas as pd
```

```
df = sns.load_dataset("titanic")
```

```
print(df.head())
```

```
   survived  pclass     sex   age  sibsp  parch     fare embarked  class  \
0         0       3    male  22.0      1      0   7.2500        S  Third   
1         1       1  female  38.0      1      0  71.2833        C  First   
2         1       3  female  26.0      0      0   7.9250        S  Third   
3         1       1  female  35.0      1      0  53.1000        S  First   
4         0       3    male  35.0      0      0   8.0500        S  Third   

     who  adult_male deck  embark_town alive  alone  
0    man        True  NaN  Southampton    no  False  
1  woman       False    C    Cherbourg   yes  False  
2  woman       False  NaN  Southampton   yes   True  
3  woman       False    C  Southampton   yes  False  
4    man        True  NaN  Southampton    no   True  
```

```
df.tail()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
886	0	2	male	27.0	0	0	13.00	S	Second	man	True	NaN	Southampton	no	True
887	1	1	female	19.0	0	0	30.00	S	First	woman	False	B	Southampton	yes	True
888	0	3	female	NaN	1	2	23.45	S	Third	woman	False	NaN	Southampton	no	False
889	1	1	male	26.0	0	0	30.00	C	First	man	True	C	Cherbourg	yes	True
890	0	3	male	32.0	0	0	7.75	Q	Third	man	True	NaN	Queenstown	no	True

```
df.sample()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
75	0	3	male	25.0	0	0	7.65	S	Third	man	True	F	Southampton	no	True

```
df.shape
```

```
(891, 15)
```

Start coding or generate with AI.

```
df.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
```

```
df['age'].mean()
```

```
np.float64(29.69911764705882)
```

```
df['age'].mode()
```

```
    age
0   24.0

dtype: float64
```

```
df['age'].median()
```

```
28.0
```

```
df['age']=df['age'].fillna(df['age'].mean())
```

```
df.isnull().sum()
```

```
                0
   survived     0
     pclass     0
        sex     0
        age     0
      sibsp     0
      parch     0
       fare     0
   embarked     2
      class     0
        who     0
 adult_male     0
embark_town     2
      alive     0
      alone     0

dtype: int64
```

```
print(df.columns)
```

```
Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare',
       'embarked', 'class', 'who', 'adult_male', 'embark_town', 'alive',
       'alone'],
      dtype='object')
```

```
df=df.drop("deck" , axis=1)
```

Show hidden output

Next steps: Explain error

```
df
```

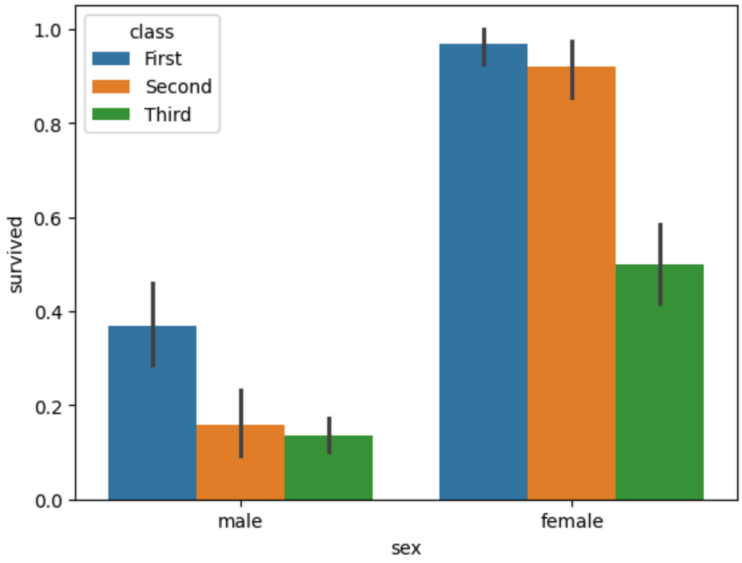
```
     survived  pclass     sex        age  sibsp  parch     fare  embarked   class    who  adult_male  embark_town  alive  alone
0           0       3    male  22.000000      1      0   7.2500         S   Third    man        True  Southampton     no  False
1           1       1  female  38.000000      1      0  71.2833         C   First  woman       False    Cherbourg    yes  False
2           1       3  female  26.000000      0      0   7.9250         S   Third  woman       False  Southampton    yes   True
3           1       1  female  35.000000      1      0  53.1000         S   First  woman       False  Southampton    yes  False
4           0       3    male  35.000000      0      0   8.0500         S   Third    man        True  Southampton     no   True
...       ...     ...     ...        ...    ...    ...      ...       ...     ...    ...         ...          ...    ...    ...
886         0       2    male  27.000000      0      0  13.0000         S  Second    man        True  Southampton     no   True
887         1       1  female  19.000000      0      0  30.0000         S   First  woman       False  Southampton    yes   True
888         0       3  female  29.699118      1      2  23.4500         S   Third  woman       False  Southampton     no  False
889         1       1    male  26.000000      0      0  30.0000         C   First    man        True    Cherbourg    yes   True
890         0       3    male  32.000000      0      0   7.7500         Q   Third    man        True   Queenstown     no   True
891 rows × 14 columns
```

Next steps: Generate code with df New interactive sheet

```
import matplotlib.pyplot as plt
```

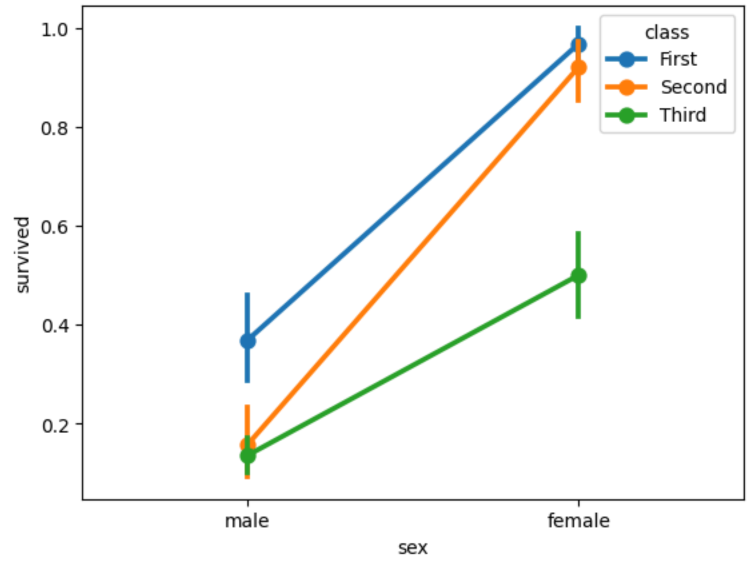
```
sns.barplot(x='sex' , y='survived' , hue='class' , data=df)
```

```
<Axes: xlabel='sex', ylabel='survived'>
```



```
sns.pointplot(x="sex", y="survived", hue="class", data=df)
```

```
<Axes: xlabel='sex', ylabel='survived'>
```

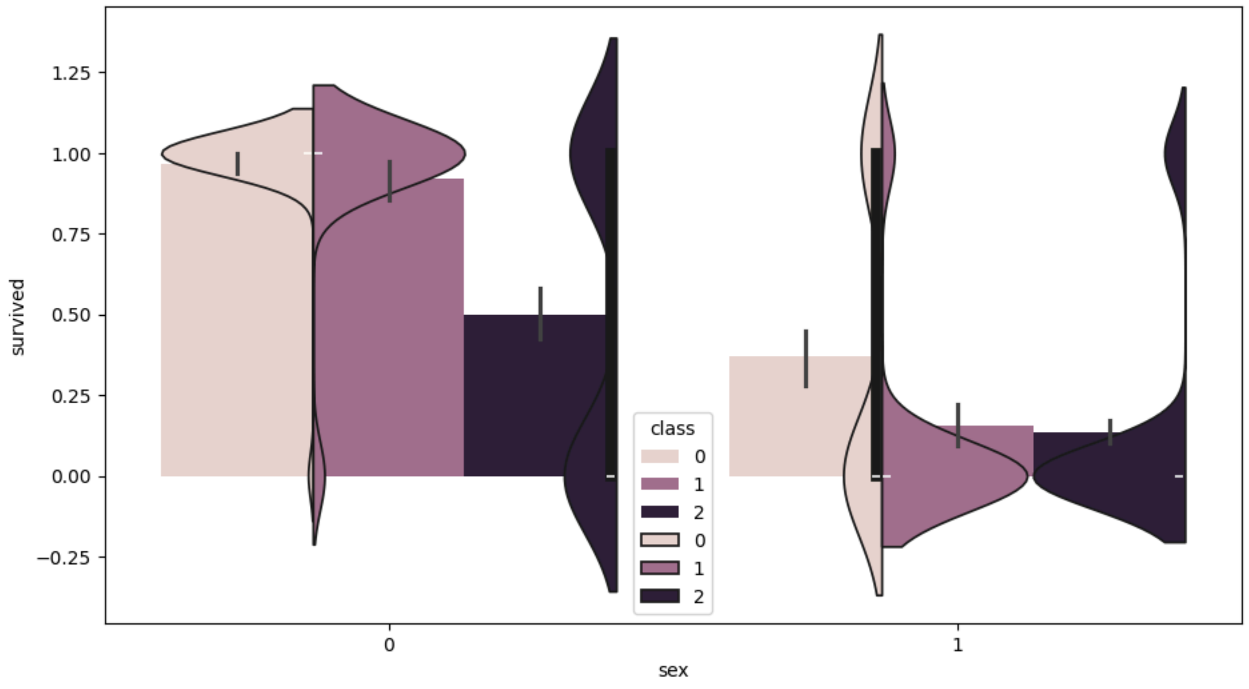


```
print(df.columns)
```

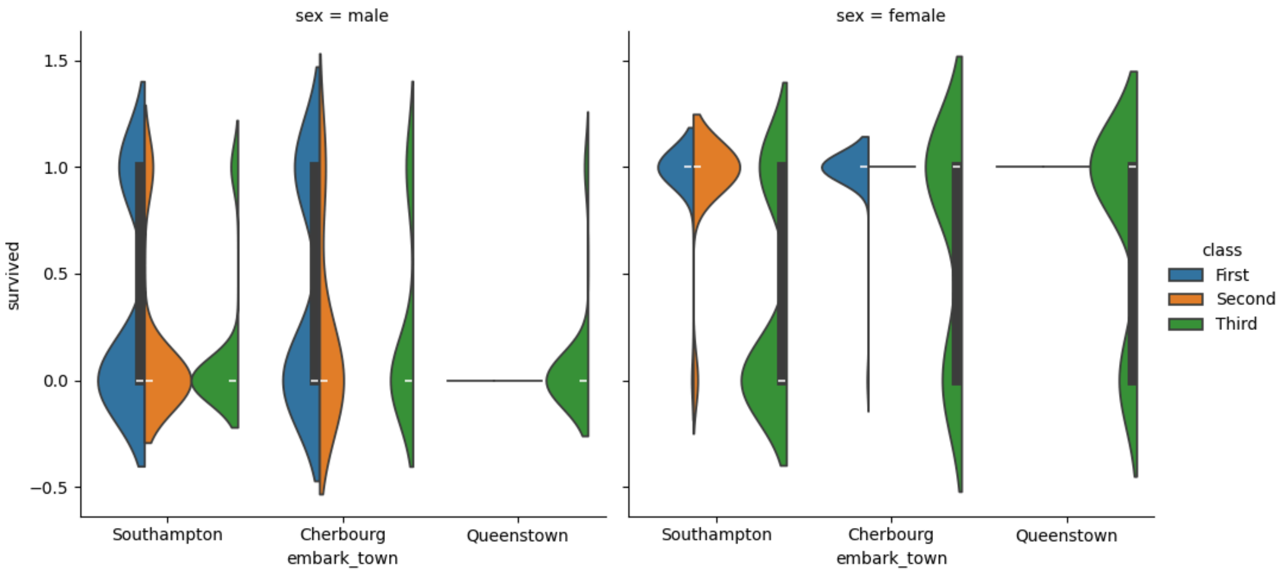
```
Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare',
       'embarked', 'class', 'who', 'adult_male', 'embark_town', 'alive',
       'alone'],
      dtype='object')
```

```
plt.figure(figsize=(11,6))
sns.barplot(x='sex' , y='survived' , hue='class' , data=df)
sns.violinplot(x="adult_male", y="survived", hue="class", data=df, split=True)
```

```
<Axes: xlabel='sex', ylabel='survived'>
```



```
g = sns.catplot(x="embark_town", y="survived", hue="class", col="sex", kind="violin", data=df, split=True)
```



```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
a=['sex' , 'embarked' , 'class' , 'who' , 'adult_male' , 'embark_town' , 'alive' , 'alone']

for i in a:
  df[i]=le.fit_transform(df[i])
```

```
df.info()
```

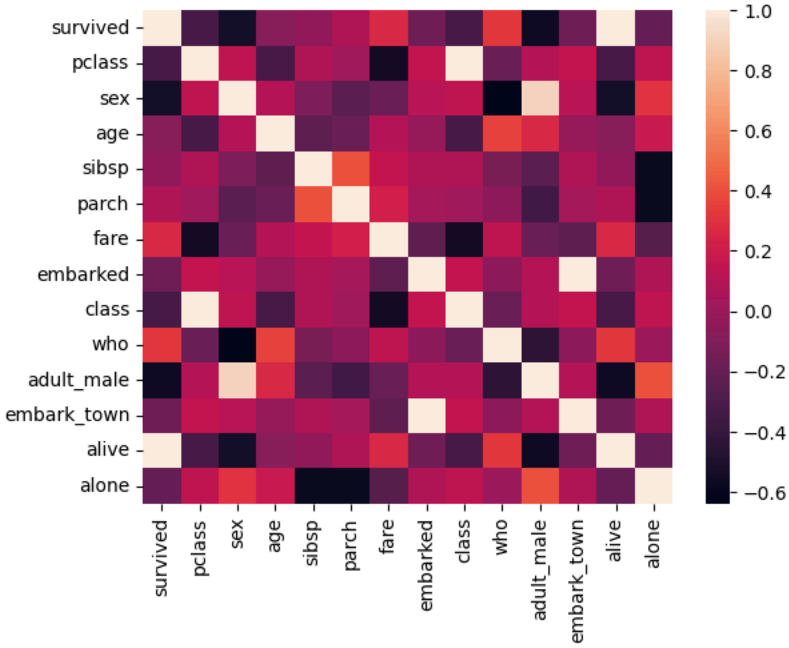
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   survived     891 non-null    int64  
 1   pclass       891 non-null    int64  
 2   sex          891 non-null    int64  
 3   age          891 non-null    float64
 4   sibsp        891 non-null    int64  
 5   parch        891 non-null    int64  
 6   fare         891 non-null    float64
 7   embarked     891 non-null    int64  
 8   class        891 non-null    int64  
 9   who          891 non-null    int64  
 10  adult_male   891 non-null    int64  
 11  embark_town  891 non-null    int64  
 12  alive        891 non-null    int64  
 13  alone        891 non-null    int64  
dtypes: float64(2), int64(12)
memory usage: 97.6 KB
```

```
df.corr()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
survived	1.000000	-0.338481	-0.543351	-0.069809	-0.035322	0.081629	0.257307	-0.163517	-0.338481	0.325753	-0.557080
pclass	-0.338481	1.000000	0.131900	-0.331339	0.083081	0.018443	-0.549500	0.157112	1.000000	-0.196793	0.094035
sex	-0.543351	0.131900	1.000000	0.084153	-0.114631	-0.245489	-0.182333	0.104057	0.131900	-0.639773	0.908578
age	-0.069809	-0.331339	0.084153	1.000000	-0.232625	-0.179191	0.091566	-0.022239	-0.331339	0.354507	0.253236
sibsp	-0.035322	0.083081	-0.114631	-0.232625	1.000000	0.414838	0.159651	0.066654	0.083081	-0.136003	-0.253586
parch	0.081629	0.018443	-0.245489	-0.179191	0.414838	1.000000	0.216225	0.038322	0.018443	-0.055682	-0.349943
fare	0.257307	-0.549500	-0.182333	0.091566	0.159651	0.216225	1.000000	-0.221226	-0.549500	0.146290	-0.182024
embarked	-0.163517	0.157112	0.104057	-0.022239	0.066654	0.038322	-0.221226	1.000000	0.157112	-0.060177	0.088725
class	-0.338481	1.000000	0.131900	-0.331339	0.083081	0.018443	-0.549500	0.157112	1.000000	-0.196793	0.094035
who	0.325753	-0.196793	-0.639773	0.354507	-0.136003	-0.055682	0.146290	-0.060177	-0.196793	1.000000	-0.437532
adult_male	-0.557080	0.094035	0.908578	0.253236	-0.253586	-0.349943	-0.182024	0.088725	0.094035	-0.437532	1.000000
embark_town	-0.163517	0.157112	0.104057	-0.022239	0.066654	0.038322	-0.221226	1.000000	0.157112	-0.060177	0.088725
alive	1.000000	-0.338481	-0.543351	-0.069809	-0.035322	0.081629	0.257307	-0.163517	-0.338481	0.325753	-0.557080
alone	-0.203367	0.135207	0.303646	0.179775	-0.584471	-0.583398	-0.271832	0.065610	0.135207	0.006540	0.404744

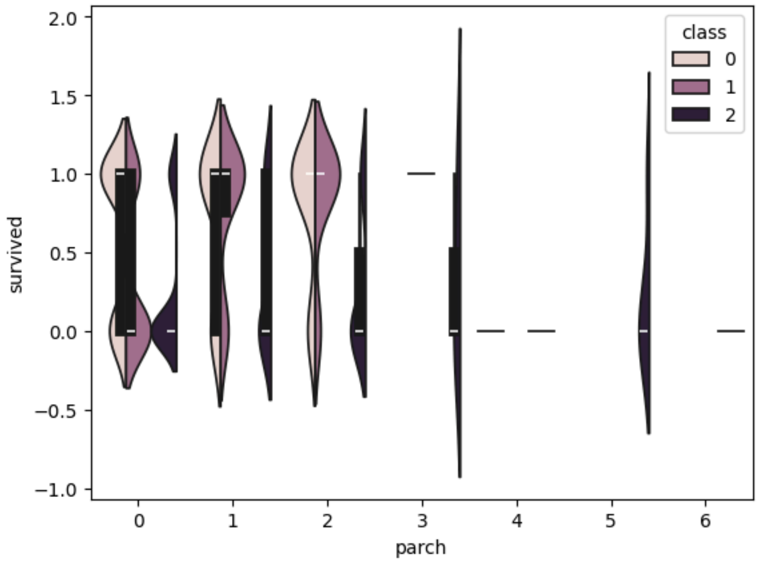
```
sns.heatmap(df.corr())
```

```
<Axes: >
```



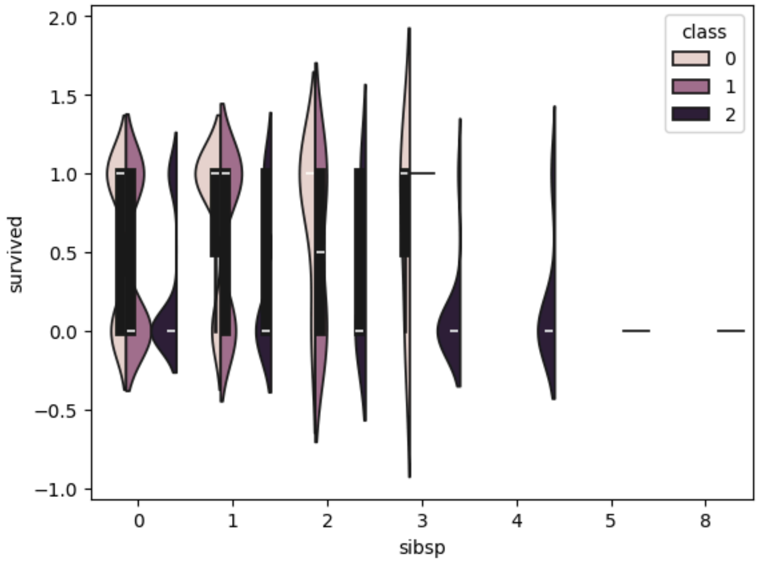
```
sns.violinplot(x="parch", y="survived", hue="class", data=df, split=True)
```

```
<Axes: xlabel='parch', ylabel='survived'>
```



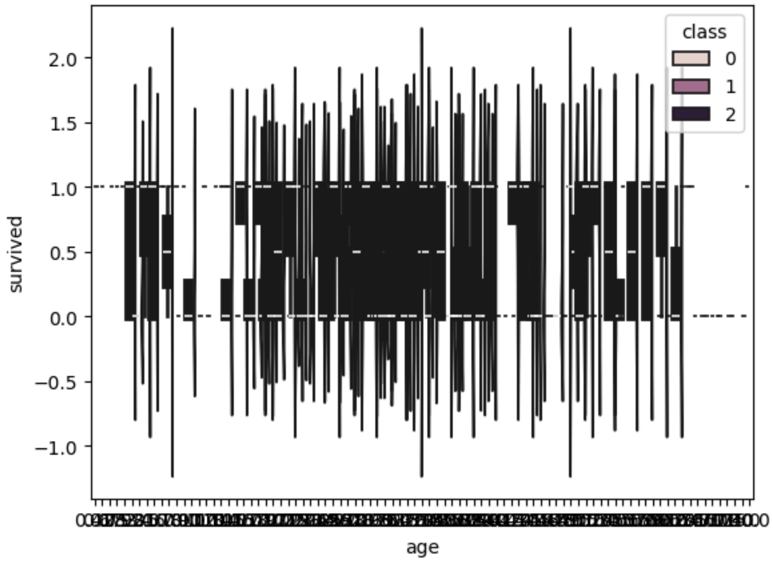
```
sns.violinplot(x="sibsp", y="survived", hue="class", data=df, split=True)
```

```
<Axes: xlabel='sibsp', ylabel='survived'>
```



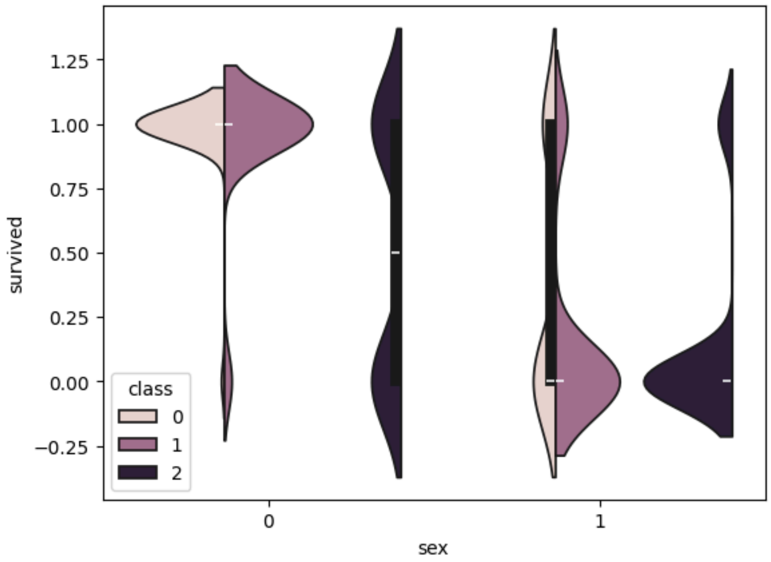
```
sns.violinplot(x="age", y="survived", hue="class", data=df, split=True)
```

```
<Axes: xlabel='age', ylabel='survived'>
```



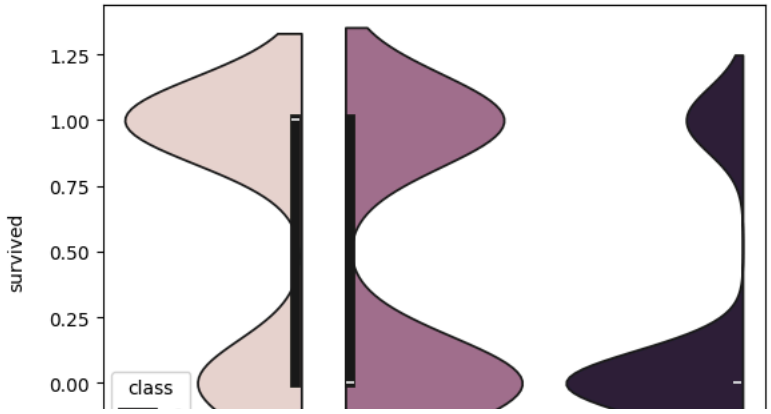
```
sns.violinplot(x="sex", y="survived", hue="class", data=df, split=True)
```

```
<Axes: xlabel='sex', ylabel='survived'>
```



```
sns.violinplot(x="pclass", y="survived", hue="class", data=df, split=True)
```

```
<Axes: xlabel='pclass', ylabel='survived'>
```


survived
class