In [1]:
```
# Importing pandas, numpy  and seaborn package
import pandas as ps
import numpy as np
import seaborn as sns
```

In [2]:
```
#Reading the dataset titanic
dataset=sns.load_dataset("titanic")
```

In [3]:
```
#dimension of dataset
dataset.shape
```

Out[3]: (891, 15)

In [4]:
```
#datatypes of dataset
dataset.dtypes
```

Out[4]:
```
survived          int64
pclass            int64
sex              object
age             float64
sibsp             int64
parch             int64
fare            float64
embarked         object
class          category
who              object
adult_male         bool
deck           category
embark_town      object
alive            object
alone              bool
dtype: object
```

In [5]:
```
#finding missing values and their sum
dataset.isnull().sum()
```

Out[5]:
```
survived        0
pclass          0
sex             0
age           177
sibsp           0
parch           0
fare            0
embarked        2
class           0
who             0
adult_male      0
deck          688
embark_town     2
alive           0
alone           0
dtype: int64
```

In [6]:
```
#printing the dataset
print(dataset)
```

```
     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
..        ...     ...     ...   ...    ...    ...      ...      ...     ...
886         0       2    male  27.0      0      0  13.0000        S  Second
887         1       1  female  19.0      0      0  30.0000        S   First
888         0       3  female   NaN      1      2  23.4500        S   Third
889         1       1    male  26.0      0      0  30.0000        C   First
890         0       3    male  32.0      0      0   7.7500        Q   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
..     ...         ...  ...          ...   ...    ...
886    man        True  NaN  Southampton    no   True
887  woman       False    B  Southampton   yes   True
888  woman       False  NaN  Southampton    no  False
889    man        True    C    Cherbourg   yes   True
890    man        True  NaN   Queenstown    no   True

[891 rows x 15 columns]
```

In [8]:
```
#drop function helps to remove the column
#removing the deck column
dataset.drop('deck',axis=1,inplace=True)
```

In [10]:
```
#initial entries are printed
#head(5) = first 5 entries are printed
dataset.head()
```

Out[10]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	embar
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	South
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	Ch
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	South
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	South
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	South

In [11]:
```
dataset.isnull().sum()
```

Out[11]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
embark_town      2
alive            0
alone            0
dtype: int64
```

In [12]:
```
#creating the copy of the dataset
x=dataset
x1=x
x1.dropna()
```

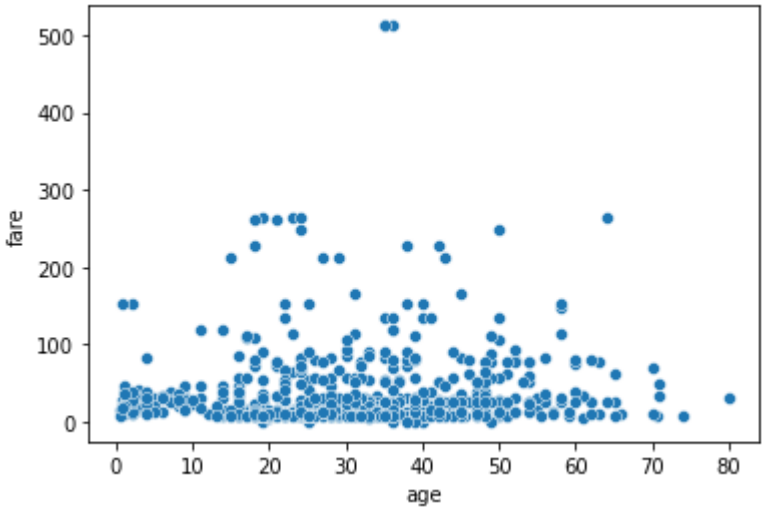
Out[12]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	en
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	S
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	S
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	S
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	S
...	...	...	...	...	...	...	...	...	...	...	...	
885	0	3	female	39.0	0	5	29.1250	Q	Third	woman	False	(
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	S
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	S
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	(

712 rows × 14 columns

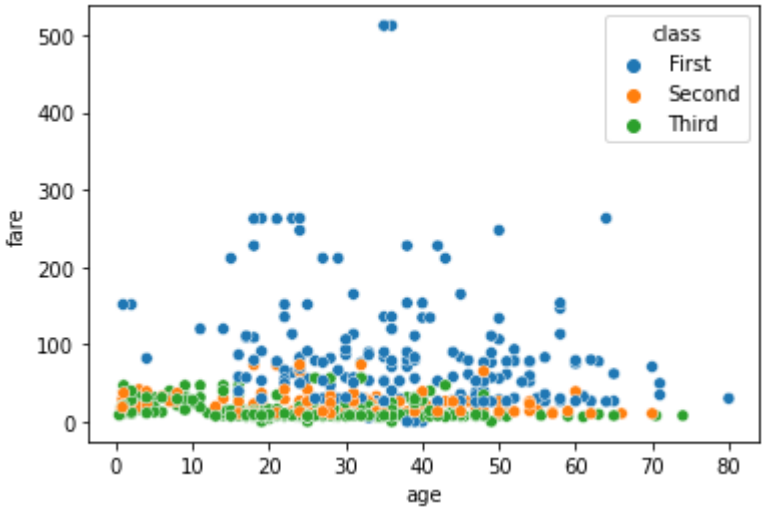
In [13]:
```
#Scatter plots are used to observe relationships between variables here variables c
sns.scatterplot(x='age',y='fare',data=x1)
```

Out[13]: <AxesSubplot:xlabel='age', ylabel='fare'>



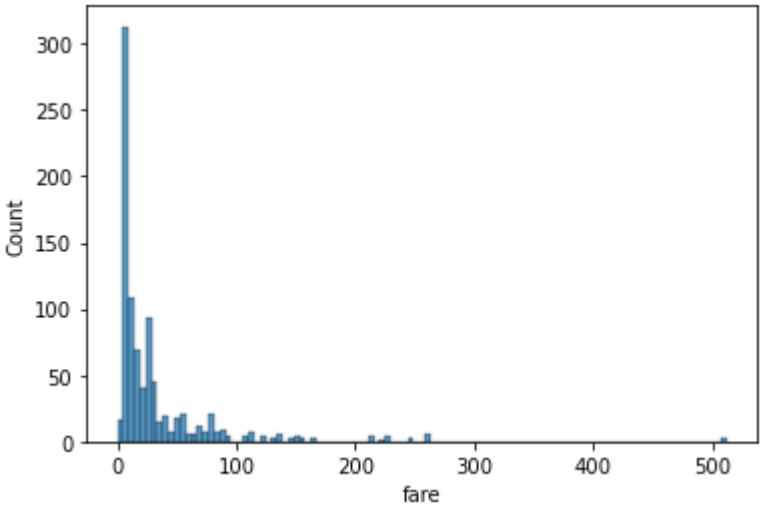
In [14]:
```
#hue -  goups variable that will produce points with different colors. e.g blue for
sns.scatterplot(x='age',y='fare',hue='class',data=x1)
```

Out[14]: <AxesSubplot:xlabel='age', ylabel='fare'>



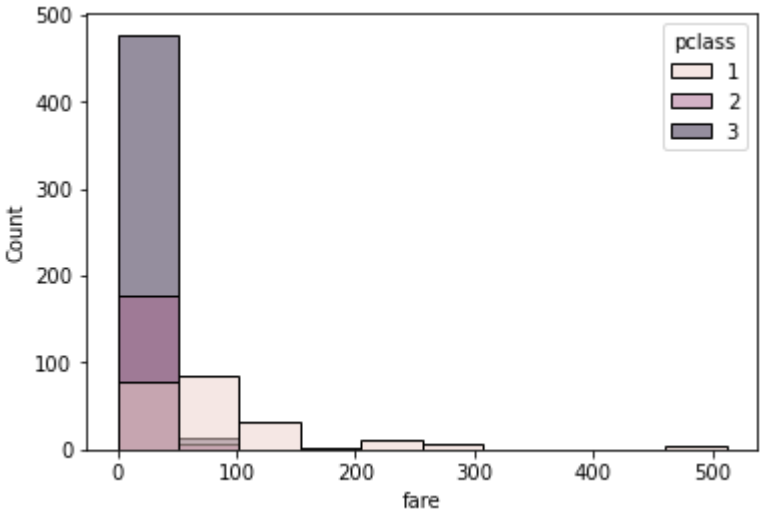
In [15]:
```
#In a histogram, each bar groups numbers into ranges.
sns.histplot(x1,x='fare')
```

Out[15]: <AxesSubplot:xlabel='fare', ylabel='Count'>



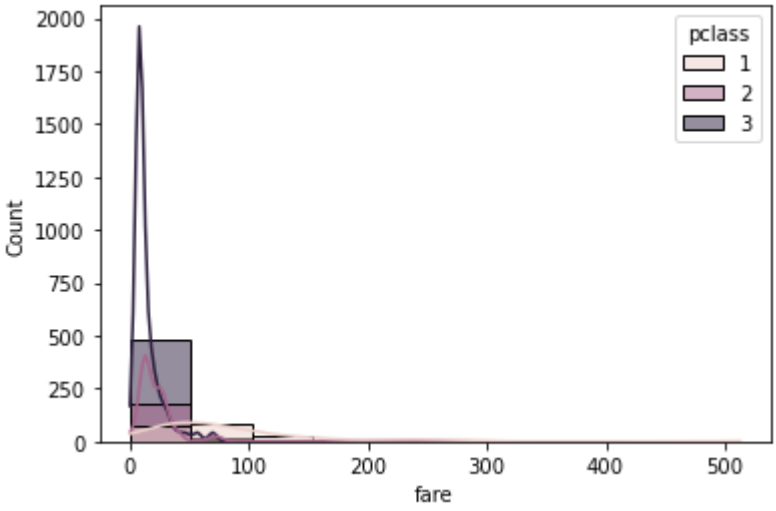
In [16]:
```
# Each bin is plotted as a bar whose height corresponds to how many data points are
sns.histplot(x1,x='fare',hue='pclass',bins=10)
```

Out[16]: <AxesSubplot:xlabel='fare', ylabel='Count'>



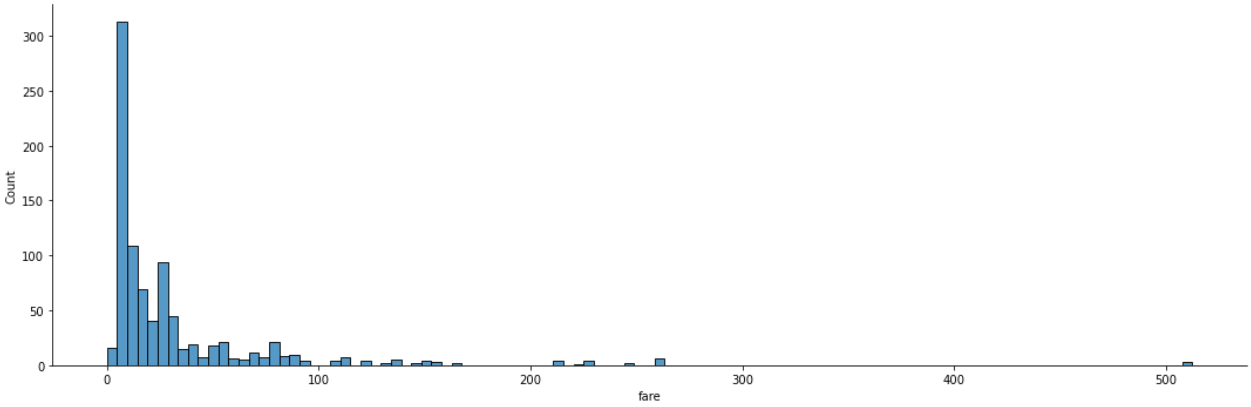
In [17]:
```
#kde - used to smooth a histogram.
sns.histplot(x1,x='fare',hue='pclass',bins=10,kde=True)
```

Out[17]: <AxesSubplot:xlabel='fare', ylabel='Count'>



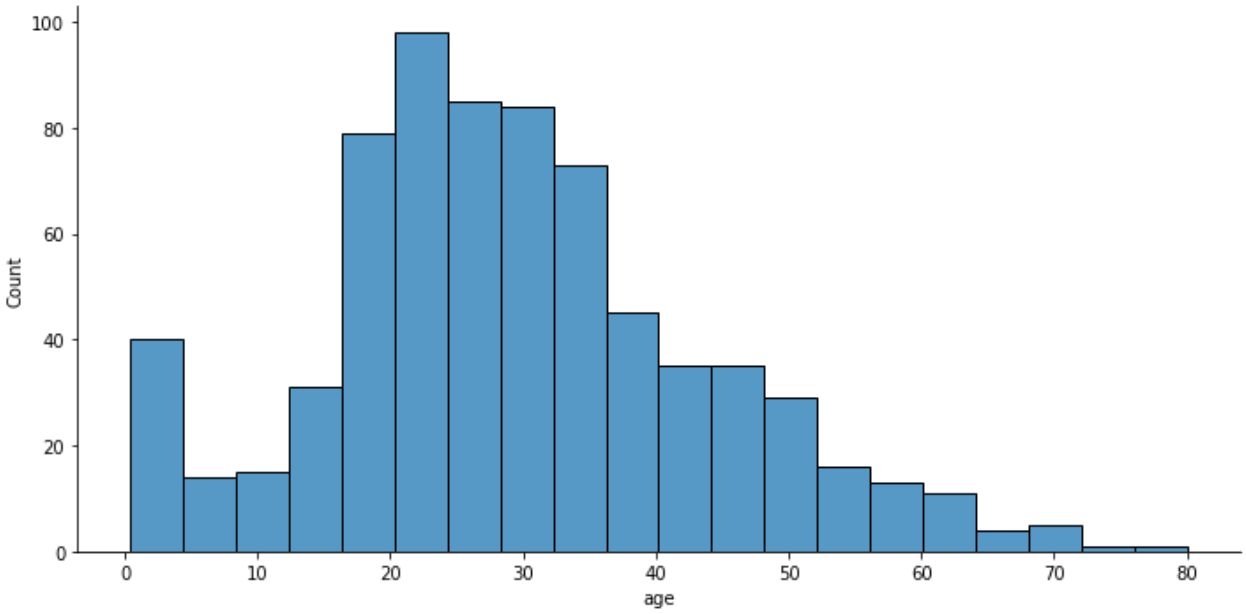
In [18]:
```
#distplot() is used to visualize the parametric distribution of a dataset.
#increase in aspect helps in broading the widthof column
sns.displot(x1['fare'],aspect=3)
```

Out[18]: <seaborn.axisgrid.FacetGrid at 0x1cdacfee460>



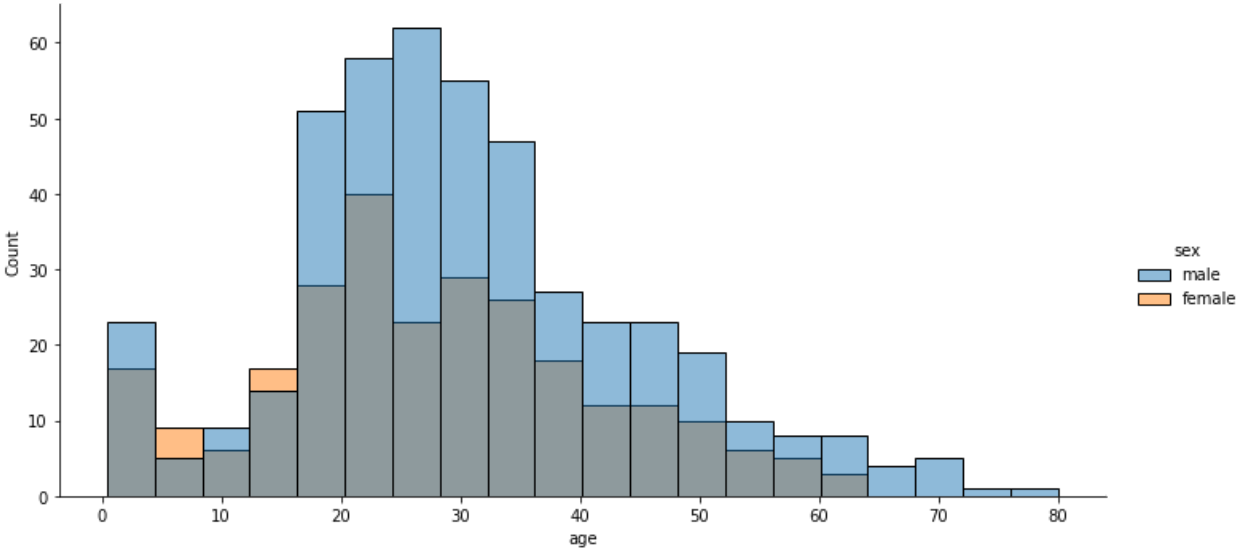
In [19]:
```
#distplot() is used to visualize the parametric distribution of a dataset.
sns.displot(x1['age'],aspect=2)
```

Out[19]: <seaborn.axisgrid.FacetGrid at 0x1cdaebb5790>



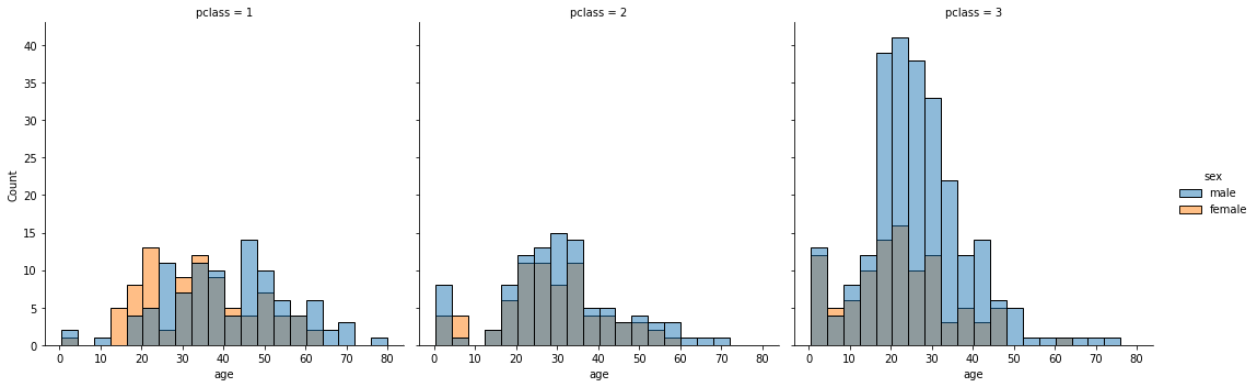
In [20]:
```
sns.displot(x1,x='age',hue='sex',aspect=2)
```

Out[20]: <seaborn.axisgrid.FacetGrid at 0x1cdaebc2f10>



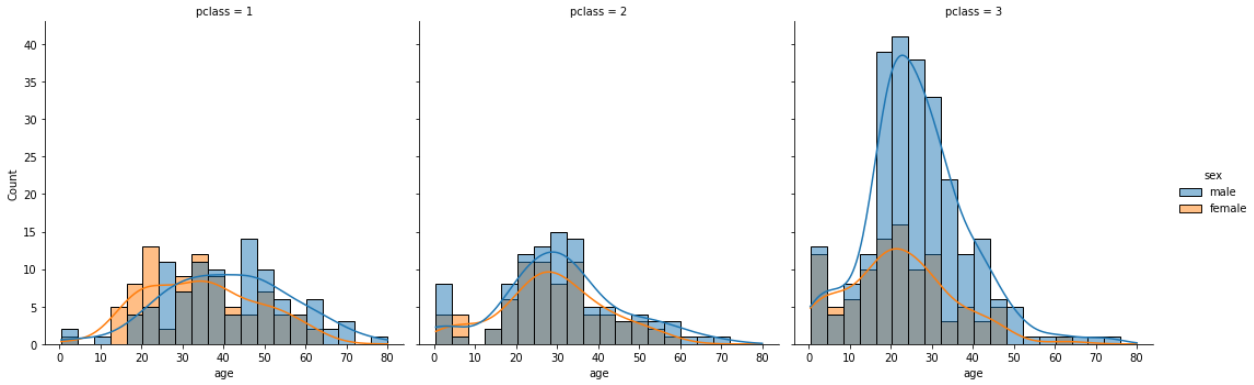
In [21]:
```
#col Variables that define subsets to plot on different facets.
sns.displot(x1,x='age',hue='sex',col='pclass')
```

Out[21]: <seaborn.axisgrid.FacetGrid at 0x1cdae4741c0>



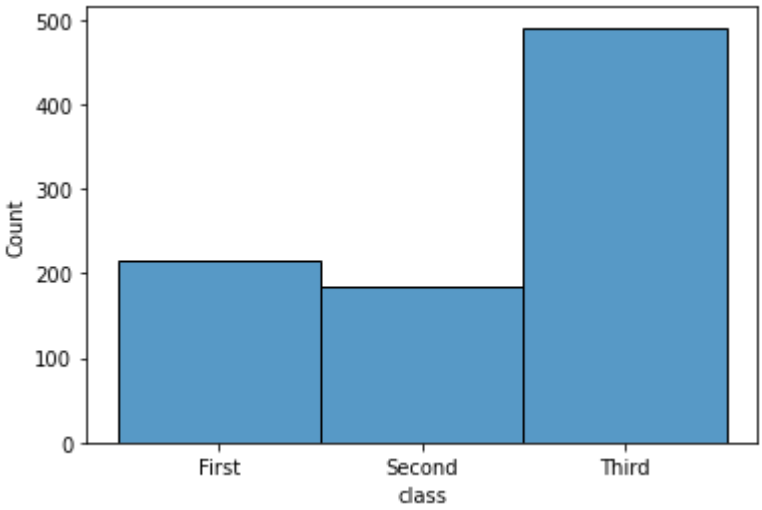
In [22]:
```
#kde - used to smooth a displot.
sns.displot(x1,x='age',hue='sex',col='pclass',kde=True)
```

Out[22]: <seaborn.axisgrid.FacetGrid at 0x1cdae474460>



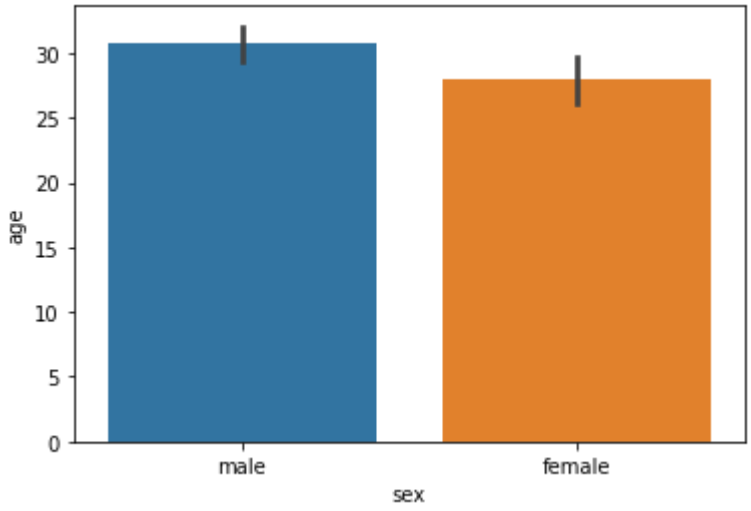
In [24]:
```
sns.histplot(x1['class'])
```

Out[24]: <AxesSubplot:xlabel='class', ylabel='Count'>



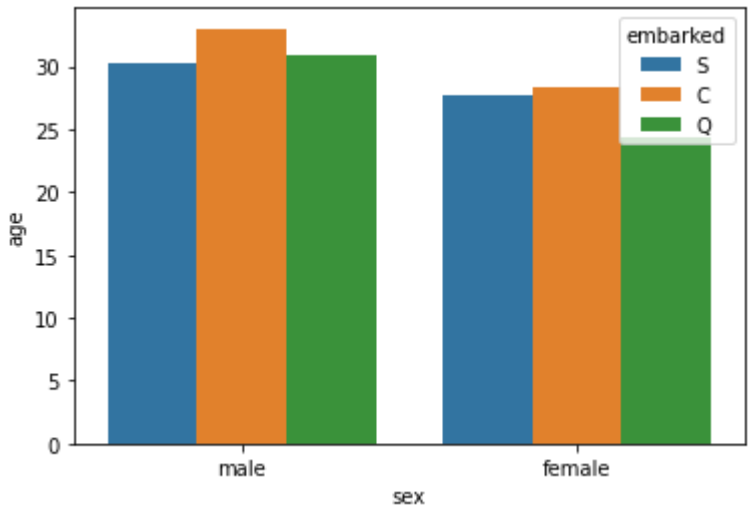
In [25]:
```
#The barplot() is used to display the mean value for each value in a categorical co
sns.barplot(x='sex',y='age',data=x1)
```

Out[25]: <AxesSubplot:xlabel='sex', ylabel='age'>



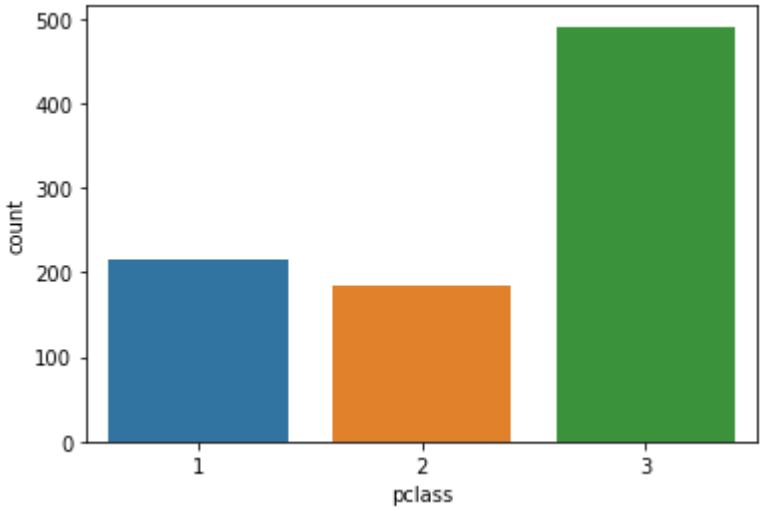
In [26]:
```
sns.barplot(x='sex',y='age',hue='embarked',data=x1,ci=None)
```

Out[26]: <AxesSubplot:xlabel='sex', ylabel='age'>



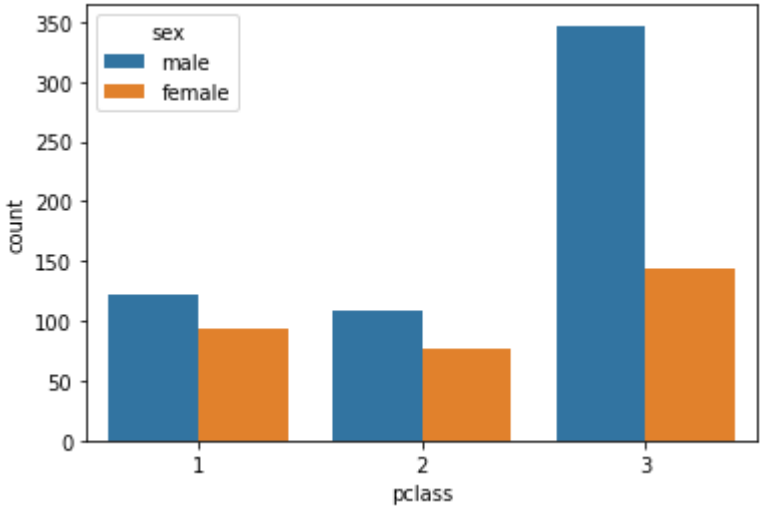
In [27]:
```
#countplot displays the count of the categories in a specific column.
sns.countplot(x='pclass',data=x1)
```

Out[27]: <AxesSubplot:xlabel='pclass', ylabel='count'>



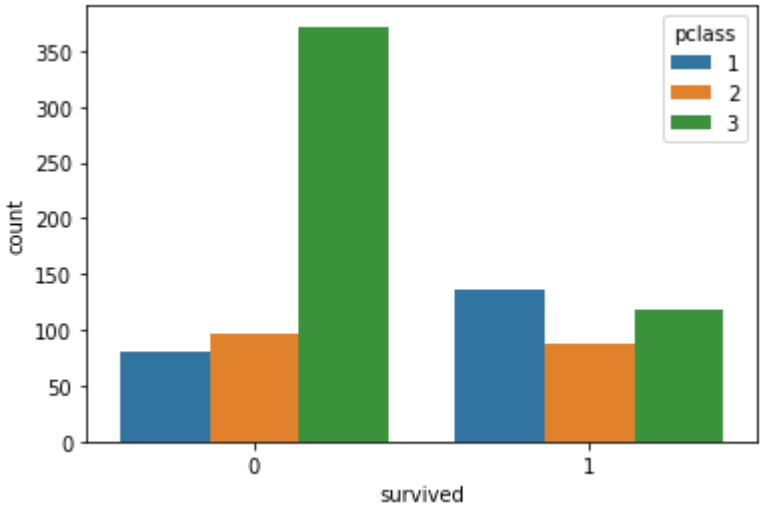
In [28]:
```
sns.countplot(x='pclass',hue='sex',data=x1)
```

Out[28]: <AxesSubplot:xlabel='pclass', ylabel='count'>



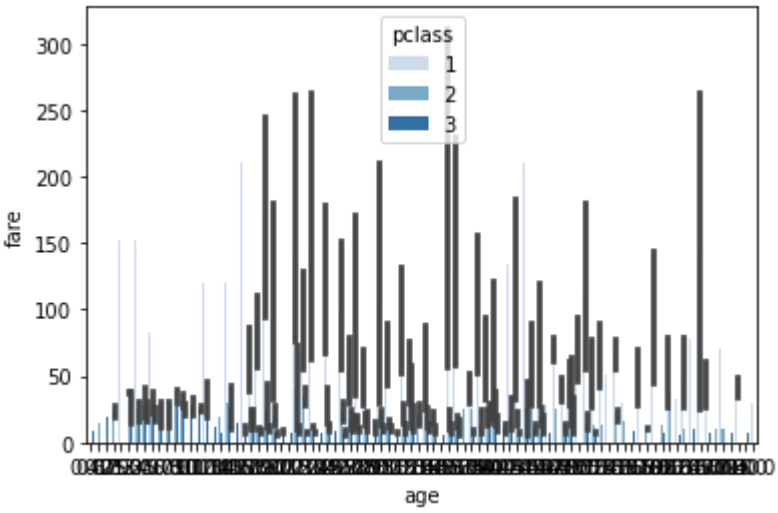
In [35]:
```
#count of people survived from classes
sns.countplot(x='survived',hue='pclass',data=x1)
```

Out[35]: <AxesSubplot:xlabel='survived', ylabel='count'>



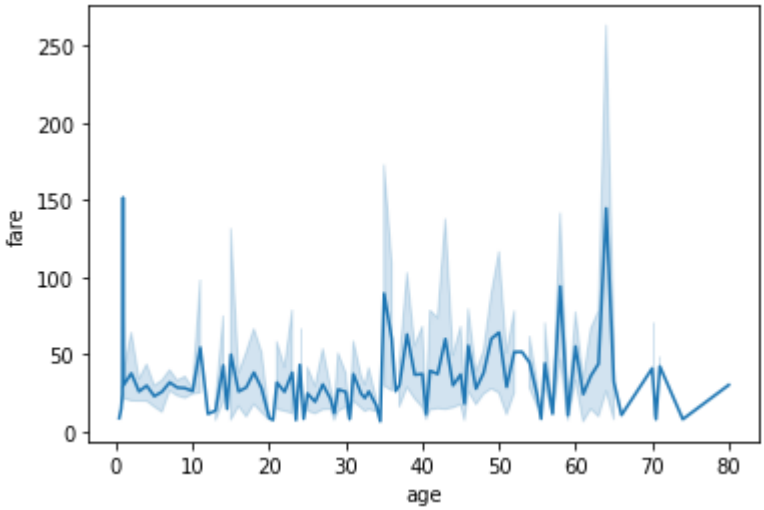
In [33]:
```
#Bar plot- to visualize passenger based on class and age along with the place from
sns.barplot(x = 'age',
            y = 'fare',
            hue = 'pclass',
            data = x1,
            palette = "Blues")
```

Out[33]: <AxesSubplot:xlabel='age', ylabel='fare'>



In [34]:
```
#lineplot is a graph that displays data as points or check marks above a number lir
#showing the frequency of each value.
sns.lineplot(x='age',y='fare',data=x1)
```

Out[34]: <AxesSubplot:xlabel='age', ylabel='fare'>



In [37]:
```
#pair plot-To plot multiple pairwise bivariate distributions in a dataset

import matplotlib.pyplot as plt
sns.pairplot(x1,hue='sex')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [37], in <cell line: 4>()
      1 #pair plot-To plot multiple pairwise bivariate distributions in a data
set
      3 import matplotlib.pyplot as plt
----> 4 sns.pairplot(x1,hue='sex')

File ~\anaconda3\lib\site-packages\seaborn\_decorators.py:46, in _deprecate_po
sitional_args.<locals>.inner_f(*args, **kwargs)
     36     warnings.warn(
     37         "Pass the following variable{} as {}keyword arg{}: {}. "
     38         "From version 0.12, the only valid positional argument "
   (...)
     43         FutureWarning
     44     )
     45 kwargs.update({k: arg for k, arg in zip(sig.parameters, args)})
---> 46 return f(**kwargs)
```

In [ ]: