

```
In [25]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

-> pandas is a data analysis library used to manipulate and analyze data in Python. It provides data structures such as DataFrame and Series for handling tabular data.

-> seaborn is a data visualization library based on matplotlib. It provides high-level interface for creating statistical graphics such as scatter plots, heatmaps, and bar plots.

-> matplotlib is a plotting library used to create static, animated, and interactive visualizations in Python. It provides a range of visualization tools for creating line plots, scatter plots, histograms, and more.

-> numpy is a numerical computing library used for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, and a range of mathematical functions for working with these arrays.

Loading "titanic" dataset

```
In [26]: df = pd.read_csv('titanic.csv')
df
```

Out[26]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	N
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	N
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	N
...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	N
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	I
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	N
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	N

891 rows × 12 columns

◀

▶

In [27]:

df.columns

Out[27]:

Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'], dtype='object')

▲

▬

▼

In [28]:

df.describe()

Out[28]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [29]: `df.isnull().sum()`

Out[29]:

```

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

```

In [30]: `df_c= df.groupby(by='Survived').agg(count=('PassengerId','count'))`
The code df.groupby(by='Survived').agg(count=('PassengerId','count')) groups the data in a D
and calculates the count of rows in each group using the 'PassengerId' column. The result is a
'count' and two rows, one for each possible value of 'Survived' (0 and 1).
`df_c`

Out[30]:

	count
Survived	
0	549
1	342

It means that in the original dataset, there were 549 passengers who did not survive (Survived=0) and 342 passengers who did survive (Survived=1).

In [31]: `sns.distplot(df['Age'])`
plot of the distribution of the 'Age' variable in the given dataset 'df', using the seaborn library.
It shows the frequency distribution of the 'Age' variable, with the x-axis representing the range
and the y-axis representing the frequency of occurrence of those ages.
The plot can give an idea of the central tendency, spread, and shape of the distribution of age.

```
C:\Users\SKY_NET\AppData\Local\Temp\ipykernel_2908\997304687.py:1: UserWarning:
```

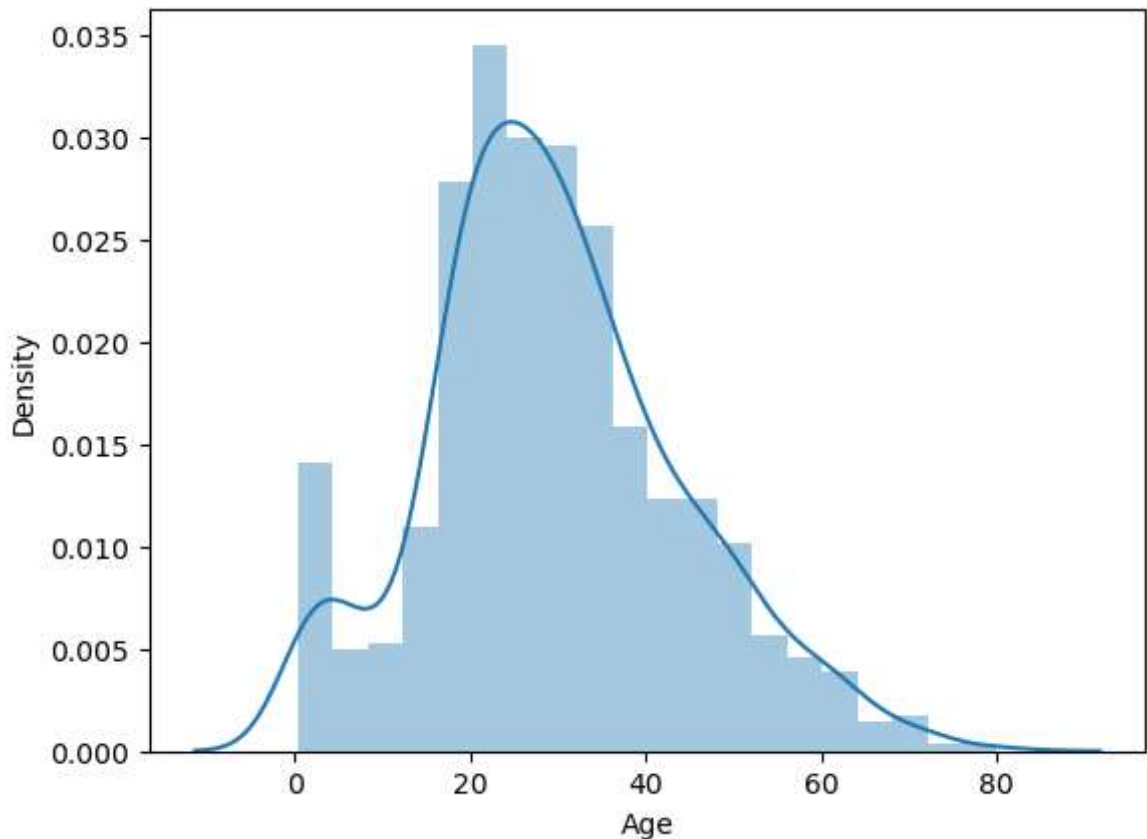
```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df['Age'])
```

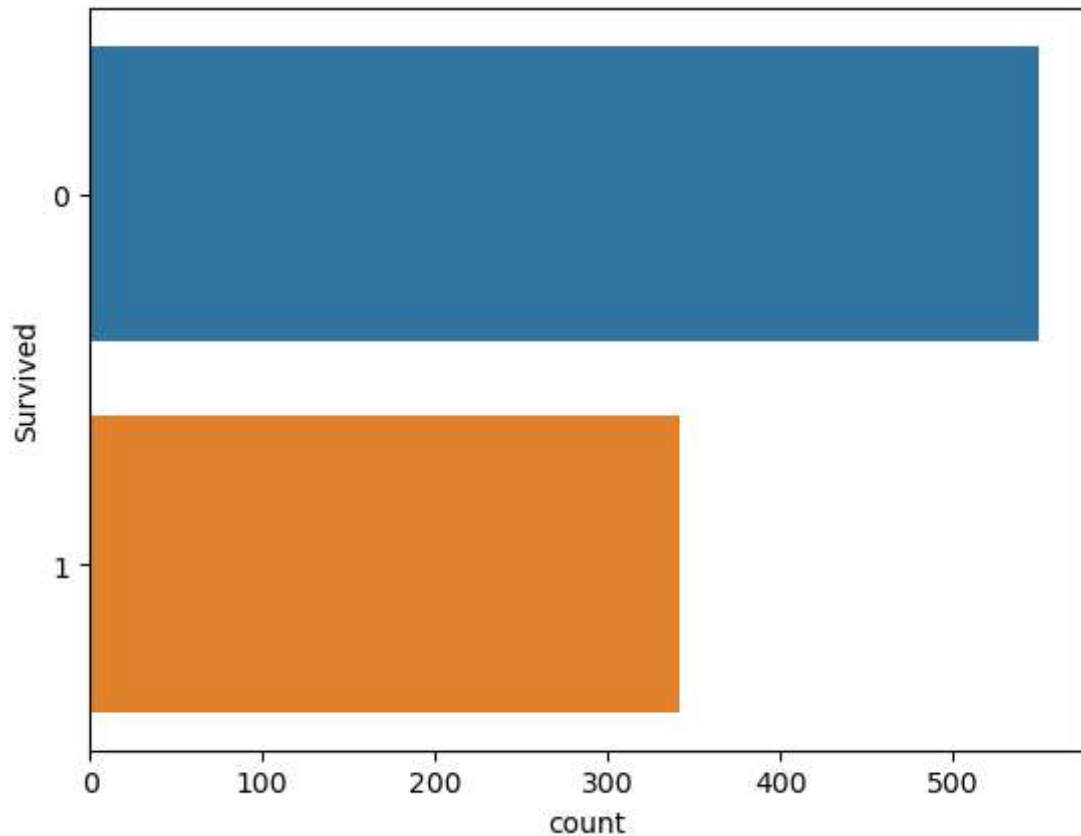
```
Out[31]: <Axes: xlabel='Age', ylabel='Density'>
```



The plot indicates there are more number of passengers of age group between 20 to 40

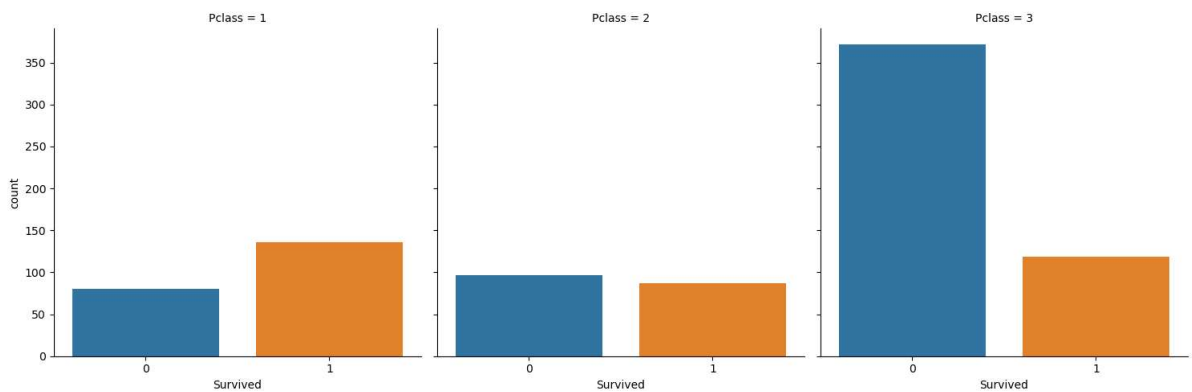
```
In [55]: sns.countplot(y="Survived", data=df)
```

```
Out[55]: <Axes: xlabel='count', ylabel='Survived'>
```



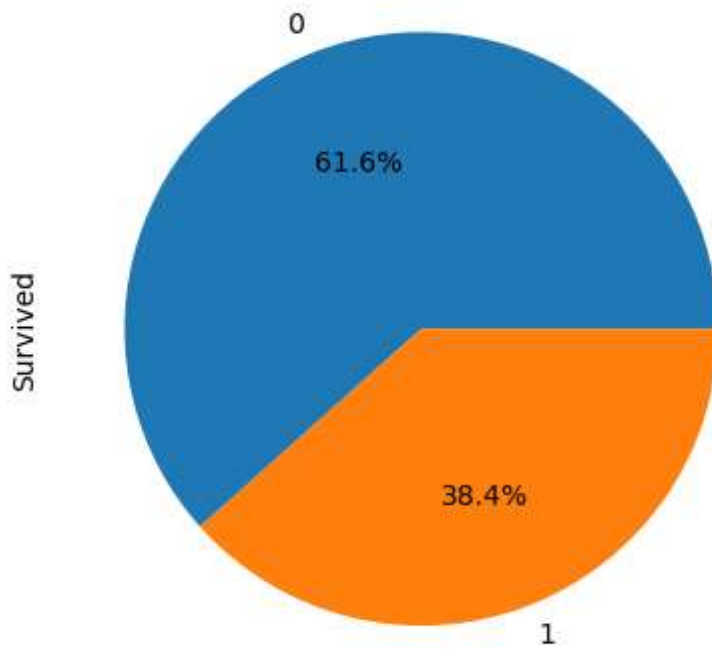
```
In [33]: sns.catplot(x="Survived",col='Pclass',  
kind="count", data = df)  
# This plot shows how many passengers survived and died for each class in the dataset,  
# giving us an idea about the survival rate among different classes.
```

Out[33]: <seaborn.axisgrid.FacetGrid at 0x269629fba90>



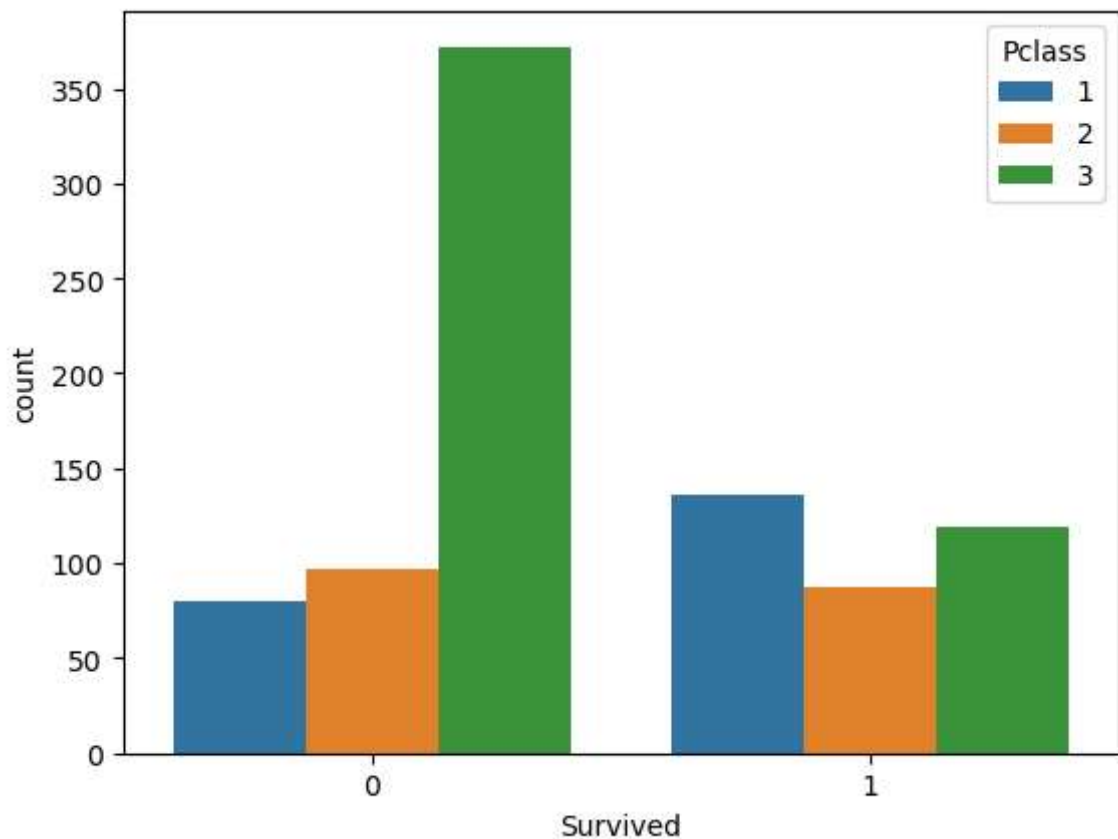
```
In [56]: df['Survived'].value_counts().plot(kind='pie', autopct='%1.1f%%')
```

Out[56]: <Axes: ylabel='Survived'>



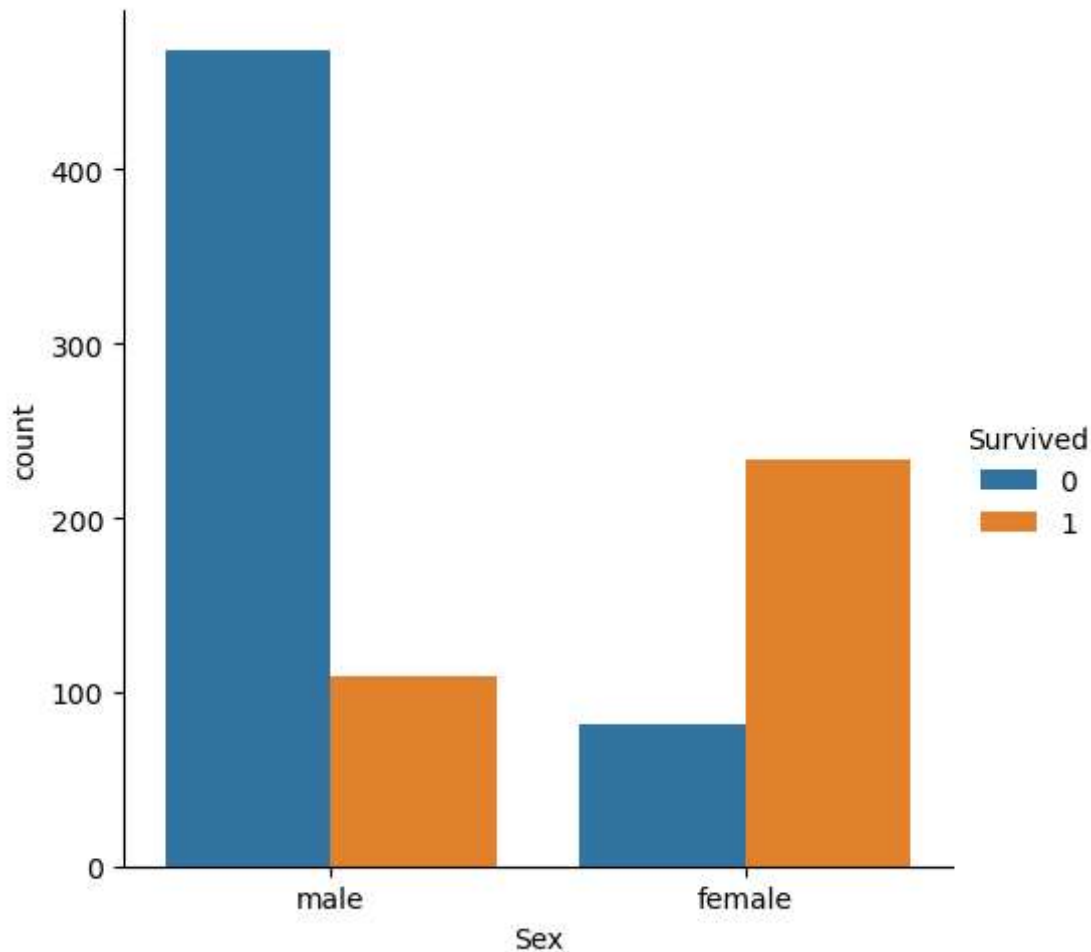
```
In [58]: sns.countplot(x='Survived', hue='Pclass', data=df)
```

```
Out[58]: <Axes: xlabel='Survived', ylabel='count'>
```

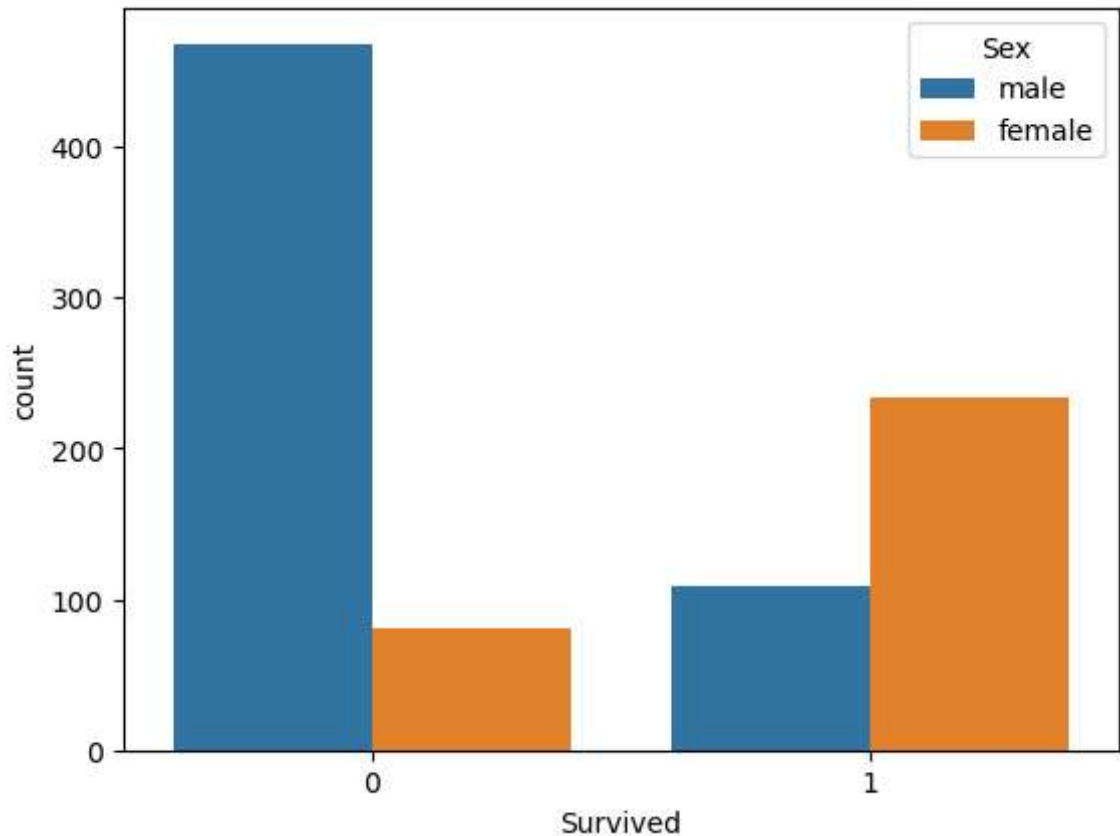


```
In [34]: sns.catplot(x="Sex", hue="Survived",  
kind="count", data=df)
```

```
Out[34]: <seaborn.axisgrid.FacetGrid at 0x26962893e50>
```



```
In [59]: sns.countplot(x="Survived", hue="Sex", data=df)
Out[59]: <Axes: xlabel='Survived', ylabel='count'>
```

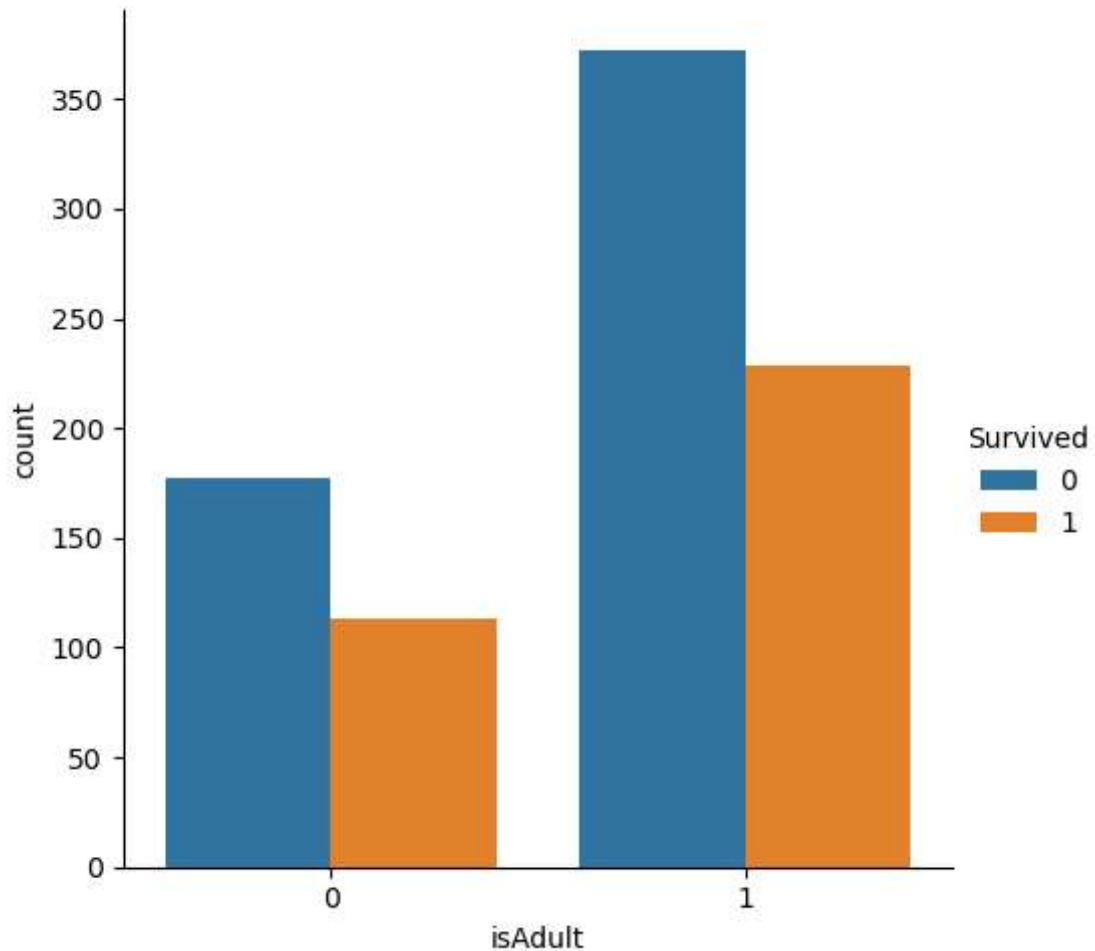


Survived vs AgeGroup (Adult / Non-Adult)

```
In [61]: df['isAdult'] = np.where(df['Age']>=18,1,0)
# creates a new column named 'isAdult' in the DataFrame 'df' based on a condition.
# If the 'Age' of a person is greater than or equal to 18, the value of 'isAdult' for that
# person is set to 1, otherwise it is set to 0.
```

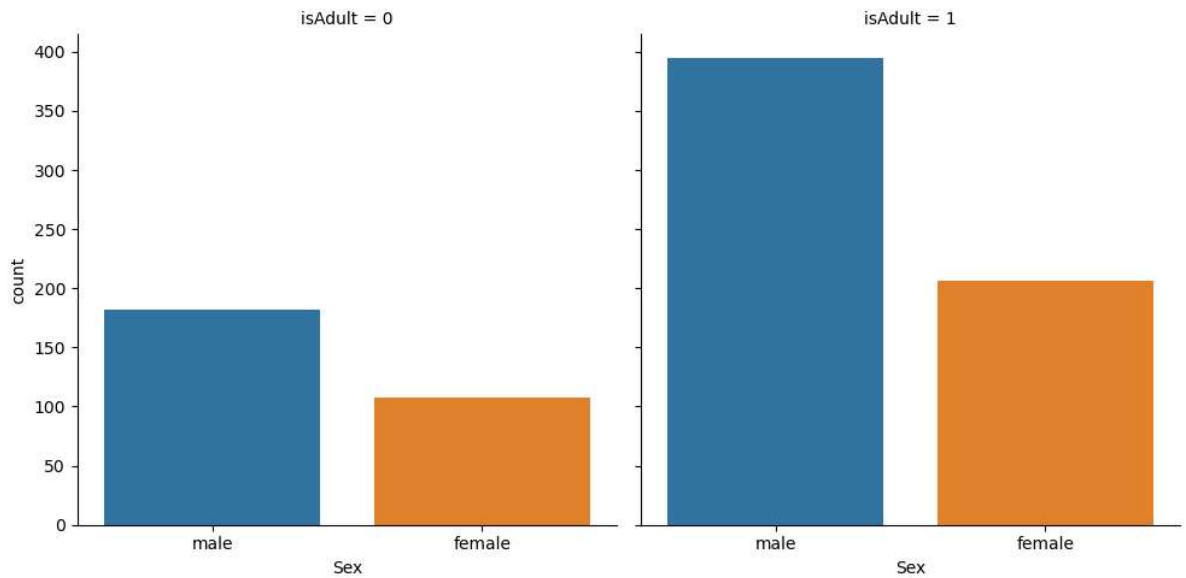
```
In [36]: sns.catplot(x="isAdult", hue="Survived",
kind="count", data=df)
```

Out[36]: <seaborn.axisgrid.FacetGrid at 0x2695d14cb20>



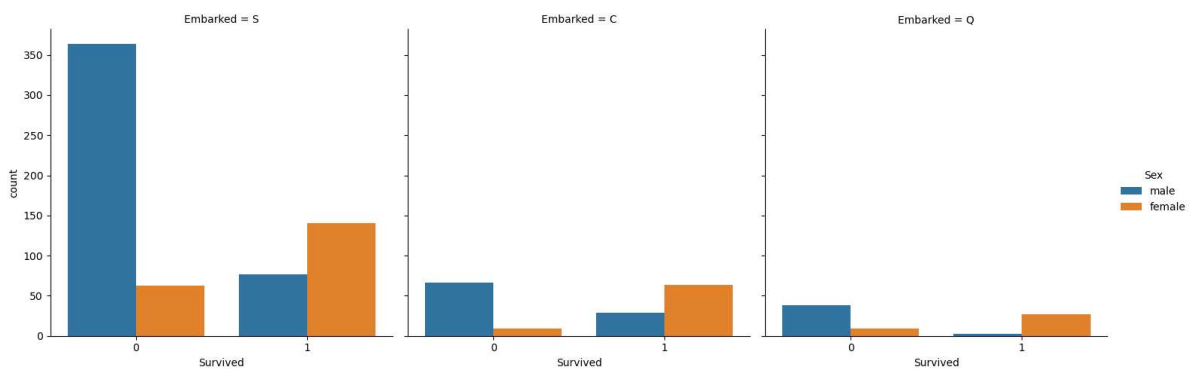
```
In [37]: sns.catplot(x="Sex",col='isAdult',
kind="count", data=df)
```

Out[37]: <seaborn.axisgrid.FacetGrid at 0x26960c47d00>



```
In [63]: sns.catplot(x="Survived",col='Embarked', hue='Sex',
kind="count", data = df)
# this plot shows the count of male and female passengers who survived and did not survive for
# It helps to visualize if there is any relationship between Embarked port, Sex, and the survival rate
```

```
Out[63]: <seaborn.axisgrid.FacetGrid at 0x2695fe48e20>
```



```
In [65]: sns.distplot(df['Fare'])
# The values on the x-axis of the plot represent the fare paid by passengers, while the values on the y-axis represent the density of the fare values.
# The plot shows the distribution of fare values in the dataset, indicating how many passengers paid a certain fare.
# The peaks in the plot indicate the fare values that were more common among the passengers.
```

C:\Users\SKY_NET\AppData\Local\Temp\ipykernel_2908\4234128465.py:1: UserWarning:

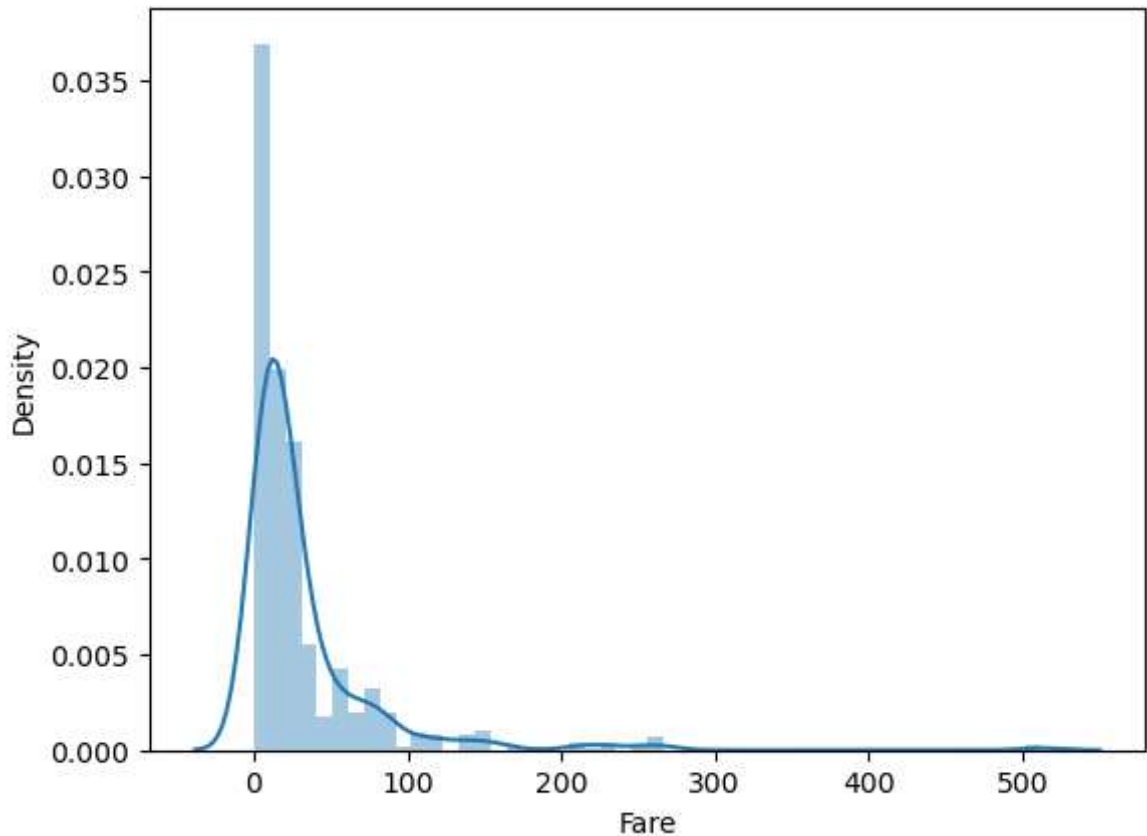
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df['Fare'])
```

```
Out[65]: <Axes: xlabel='Fare', ylabel='Density'>
```



In [40]: `df.corr()`

C:\Users\SKY_NET\AppData\Local\Temp\ipykernel_2908\1134722465.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
`df.corr()`

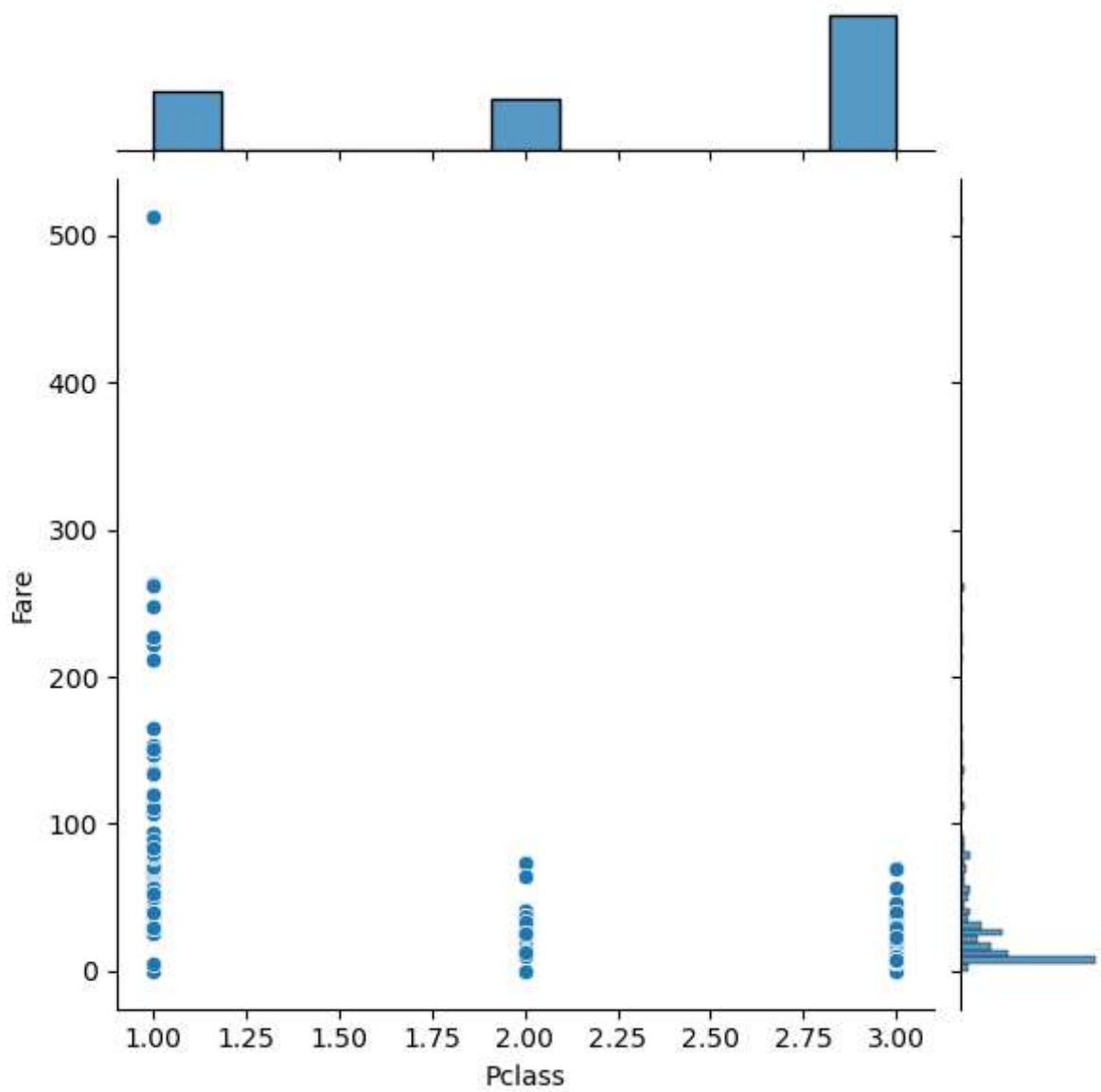
Out[40]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	isAdult
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658	0.034839
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307	-0.008309
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500	-0.236475
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067	0.617063
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651	-0.246303
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225	-0.119937
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000	0.091114
isAdult	0.034839	-0.008309	-0.236475	0.617063	-0.246303	-0.119937	0.091114	1.000000

In [66]: `sns.jointplot(x = "Pclass", y = "Fare",
kind = "scatter", data = df)`

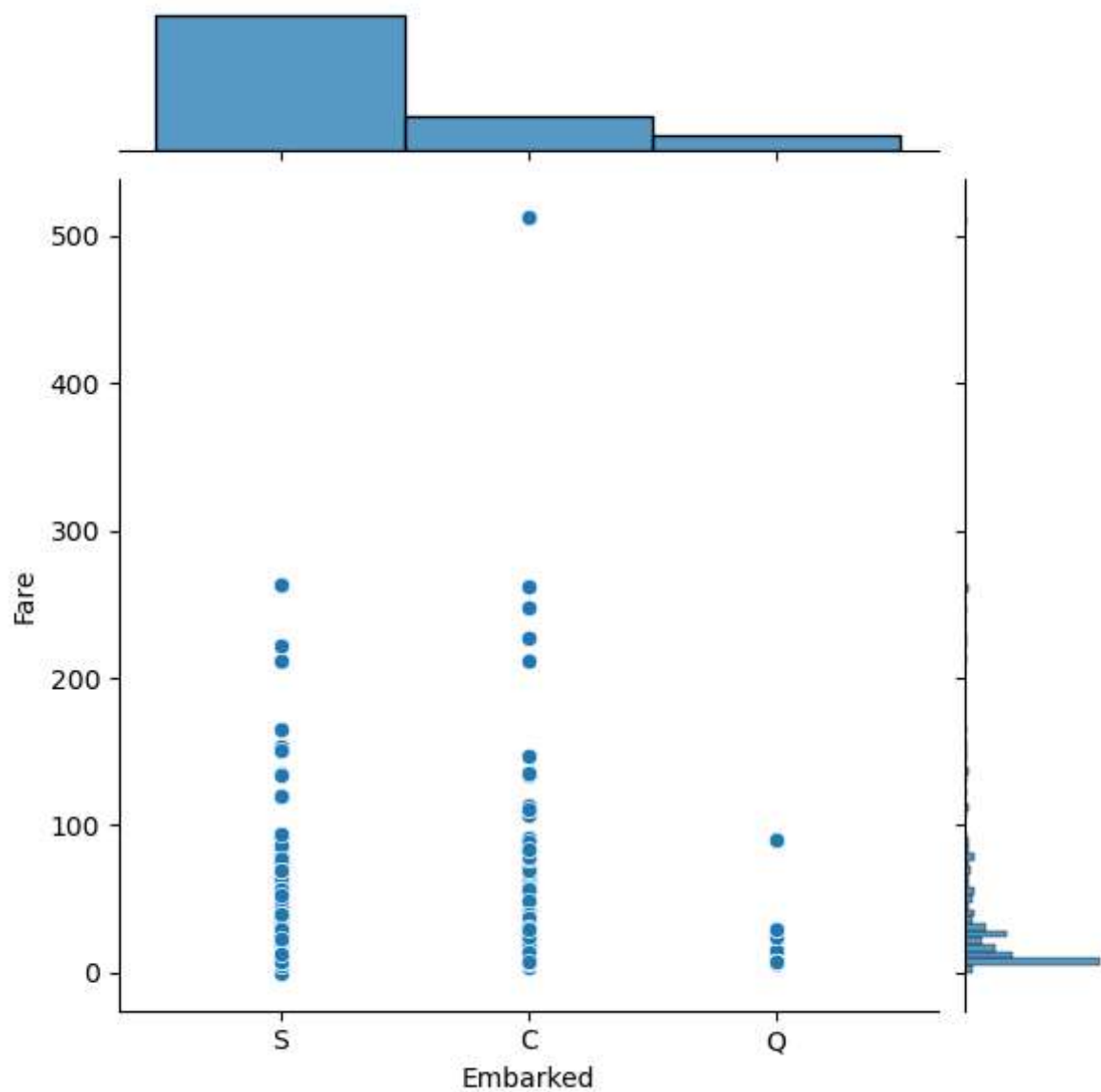
*# This plot shows the relationship between the two variables 'Pclass' and 'Fare'. The x-axis represents the passenger class and the y-axis represents the fare they paid.
Each point on the plot represents a passenger in the dataset. The joint plot displays both a scatter plot and histograms for each variable.
The scatter plot shows the individual data points while the histograms show the distribution of each variable.
This plot helps to visualize the relationship between the passenger class and fare they paid.*

Out[66]: `<seaborn.axisgrid.JointGrid at 0x2696717f1c0>`



```
In [42]: sns.jointplot(x = "Embarked", y = "Fare",  
                      kind = "scatter", data = df)
```

```
Out[42]: <seaborn.axisgrid.JointGrid at 0x26963f908b0>
```



```
In [43]: sns.jointplot(x = "Age", y = "Fare",  
                      kind = "scatter", data = df)
```

```
Out[43]: <seaborn.axisgrid.JointGrid at 0x2696467ec50>
```

