

Experiment No. 8

Aim: Use the inbuilt dataset “titanic” the dataset contains 891 rows and contains information about the passengers who boarded the unfortunate titanic ship use the seaborn library to see if we can find any patterns in the data.

Requirement:

- Anaconda Installer
- Windows 10 OS
- Linux
- Jupyter Notebook

Theory:

What is Data Visualization?

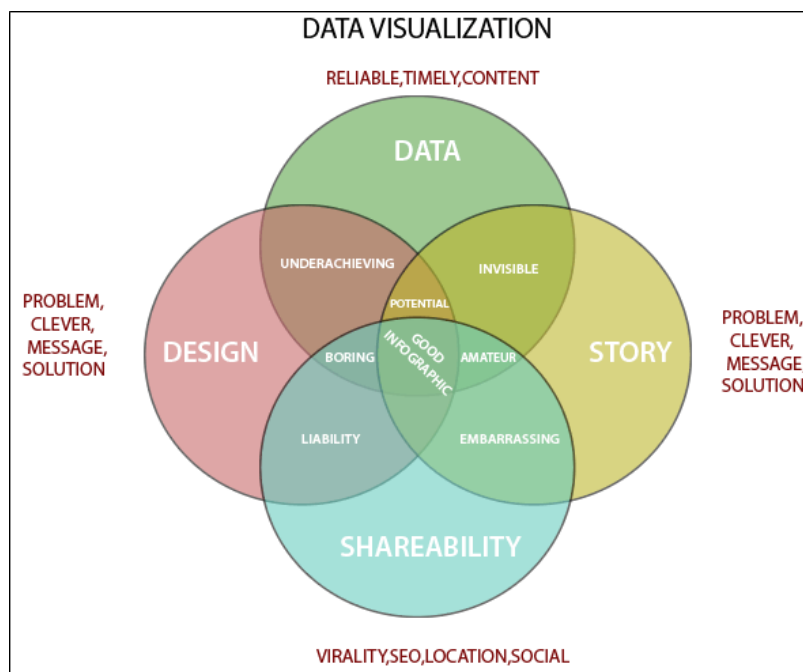


Fig. Data Visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

The benefits of data visualization:

When considering business strategies and goals, data visualization benefits decision makers in several ways to improve data insights. Let's explore seven major benefits in detail:

- Better analysis
- Quick action
- Identifying patterns
- Finding errors
- Understanding the story
- Exploring business insights
- Grasping the Latest Trends

Better analysis:

Data visualization helps business stakeholders analyze reports regarding sales, marketing strategies, and product interest. Based on the analysis, they can focus on the areas that require attention to increase profits, which in turn makes the business more productive.

Quick action:

As mentioned previously, the human brain grasps visuals more easily than table reports. Data visualizations allow decision makers to be notified quickly of new data insights and take necessary actions for business growth.

Identifying patterns:

Large amounts of complicated data can provide many opportunities for insights when we visualize them. Visualization allows business users to recognize relationships between the data, providing greater meaning to it. Exploring these patterns helps users focus on specific areas that require attention in the data, so that they can identify the significance of those areas to drive their business forward.

Finding errors:

Visualizing your data helps quickly identify any errors in the data. If the data tends to suggest the wrong actions, visualizations help identify erroneous data sooner so that it can be removed from analysis.

Understanding the story:

Storytelling is the purpose of your dashboard. By designing your visuals in a meaningful way, you help the target audience grasp the story in a single glance. Always be sure to convey the story in the simplest way, without excessive complicated visuals.

Exploring business insights:

In the current competitive business environment, finding data correlations using visual representations is key to identifying business insights. Exploring these insights is important for business users or executives to set the right path to achieving the business' goals.

Grasping the latest trends:

Using data visualization, you can discover the latest trends in your business to provide quality products and identify problems before they arise. Staying on top of trends, you can put more effort into increasing profits for your business.

How data visualization works:

Data visualization involves handling tons of data that will be converted into meaningful visuals using widgets. To achieve this, we require the best software tools to operate various types of data sources such as files, web API data, database-maintained sources, and others. Organizations should choose the best data visualization tool to meet all their requirements.

At a minimum, the tool should support interactive visual creation, flexible connectivity to data sources, combining data sources, automatic refresh of data, sharing visuals with others, secured access to data sources, and exporting widgets. These features allow you to make the best visuals of your data and also save your business time.

Trusted, real-time data visualization:

You need a way to quickly pivot your company's efforts in response to world and customer-evolving expectations. You also need a way to make these quick business decisions using big data. But big data has been increasing in volume-becoming even bigger data. As a result, the massive amount of data is slow to sort through, comprehend and especially explain. And, if you can pull outcomes from disparate sources, it isn't easy to interpret their numerical outputs.

Libraries Used:

1. Seaborn: Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python.

Conclusion:

In this experiment we have studied about what is data visualization. How data visualization helps us in different ways. We have also used the inbuilt data set i.e., Titanic from the seaborn library to perform data visualization on it.

```
In [2]: import seaborn as sns
df= sns.load_dataset('titanic')
```

```
In [3]: df
```

```
Out[3]:
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True
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
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True
...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False
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

891 rows × 15 columns



```
In [4]: df=df[['survived','class','sex','age','fare']]
```

```
In [5]: df
```

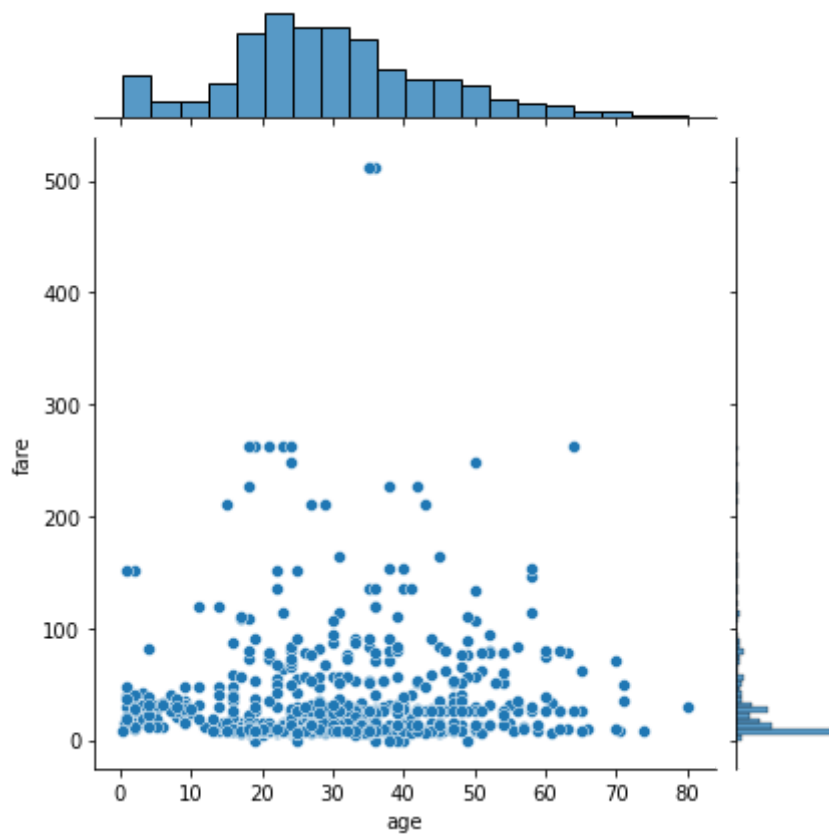
```
Out[5]:
```

	survived	class	sex	age	fare
0	0	Third	male	22.0	7.2500
1	1	First	female	38.0	71.2833
2	1	Third	female	26.0	7.9250
3	1	First	female	35.0	53.1000
4	0	Third	male	35.0	8.0500
...
886	0	Second	male	27.0	13.0000
887	1	First	female	19.0	30.0000
888	0	Third	female	NaN	23.4500
889	1	First	male	26.0	30.0000
890	0	Third	male	32.0	7.7500

891 rows × 5 columns

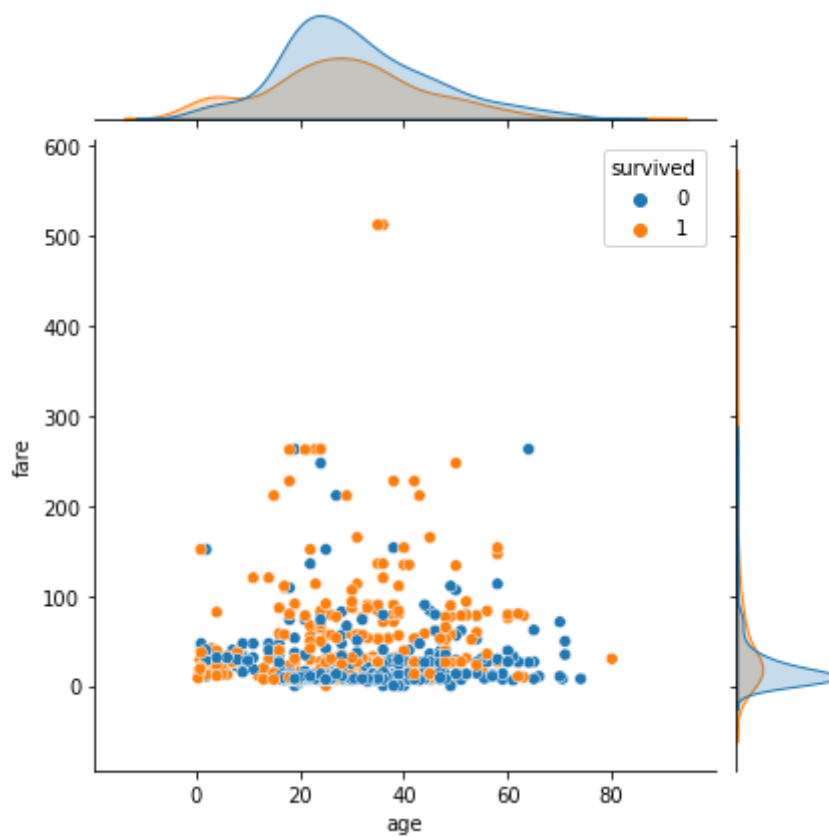
```
In [6]: sns.jointplot(x='age',y='fare',data=df)
```

```
Out[6]: <seaborn.axisgrid.JointGrid at 0x1e32202c700>
```



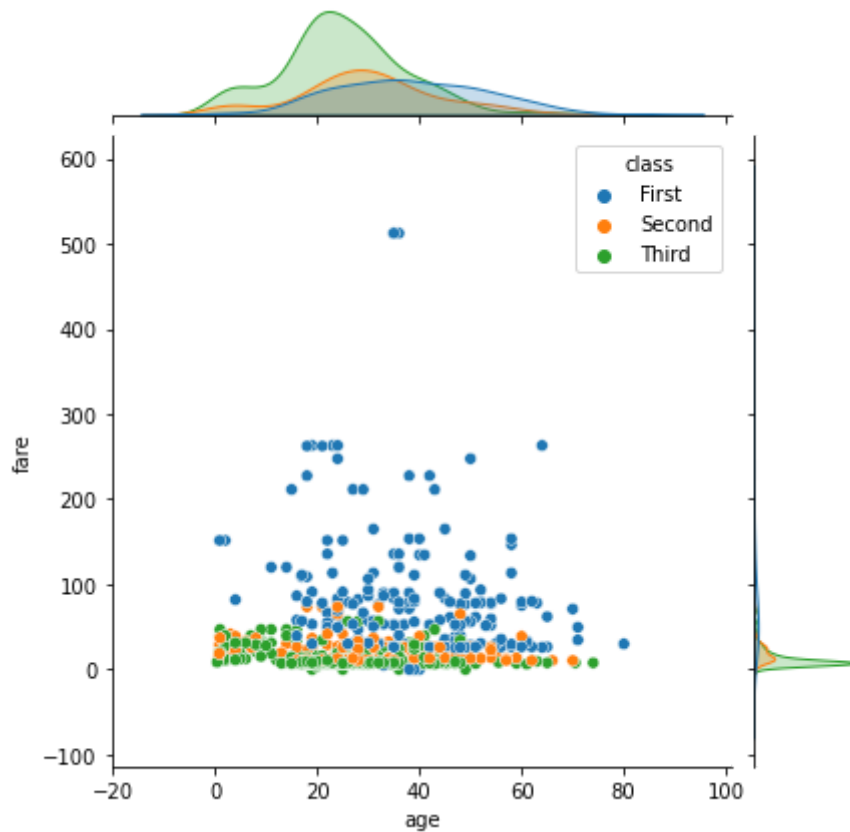
```
In [7]: sns.jointplot(x='age',y='fare',data=df,hue='survived')
```

```
Out[7]: <seaborn.axisgrid.JointGrid at 0x1e322a5dfd0>
```



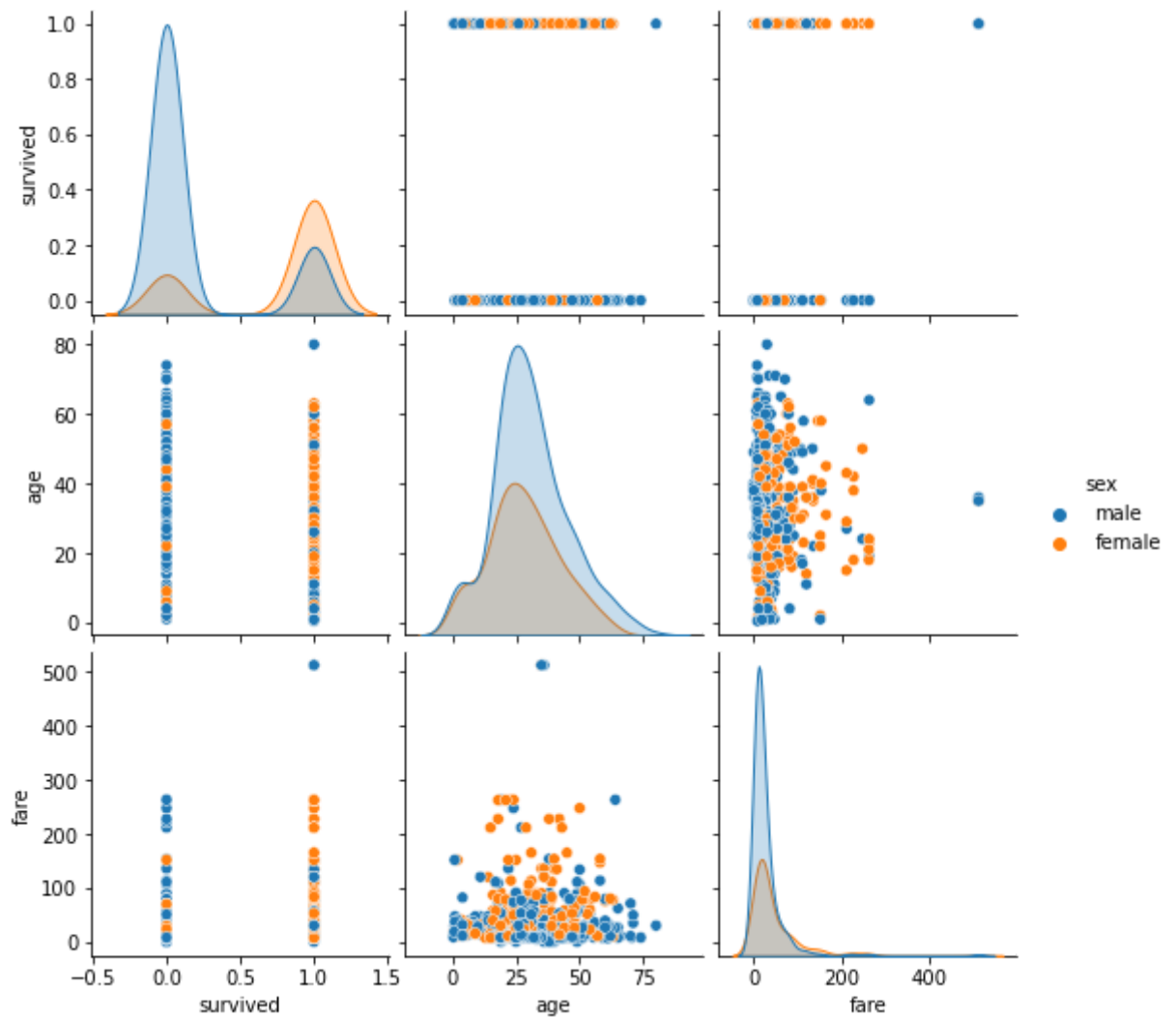
```
In [8]: sns.jointplot(x='age',y='fare',data=df,hue='class')
```

```
Out[8]: <seaborn.axisgrid.JointGrid at 0x1e322b88550>
```



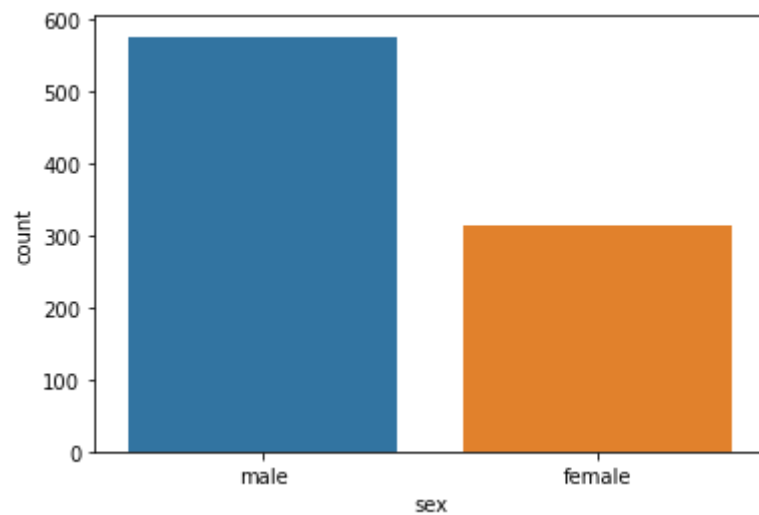
```
In [9]: sns.pairplot(df,hue='sex')
```

```
Out[9]: <seaborn.axisgrid.PairGrid at 0x1e322c6b7c0>
```



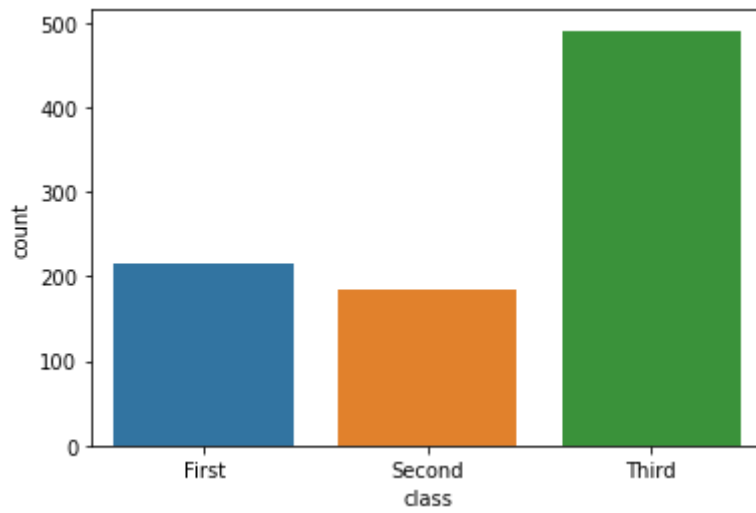
```
In [10]: sns.countplot(x=df['sex'])
```

```
Out[10]: <AxesSubplot:xlabel='sex', ylabel='count'>
```



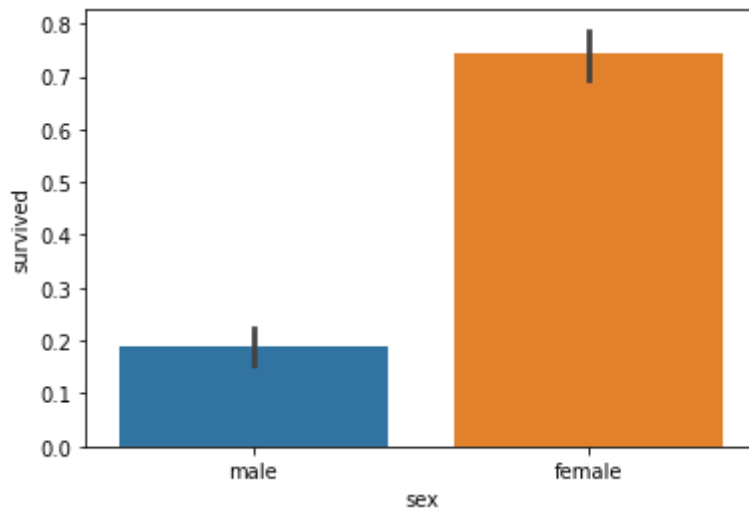
```
In [11]: sns.countplot(x=df['class'])
```

```
Out[11]: <AxesSubplot:xlabel='class', ylabel='count'>
```



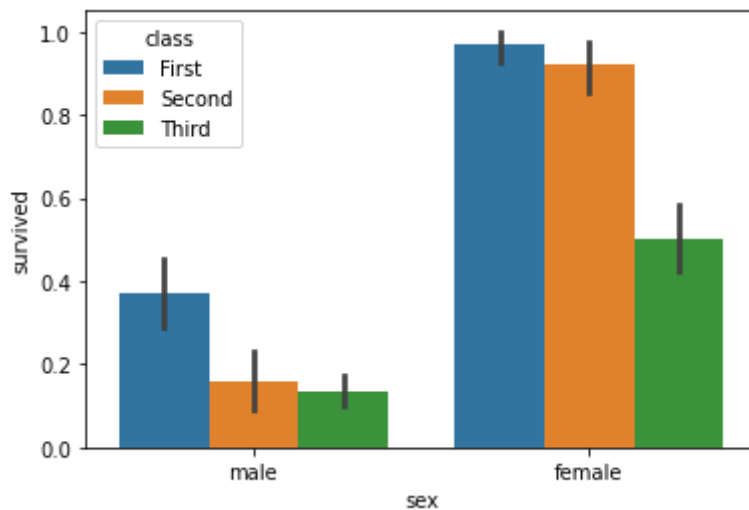
In [12]: `sns.barplot(x='sex',y='survived',data=df)`

Out[12]: `<AxesSubplot:xlabel='sex', ylabel='survived'>`



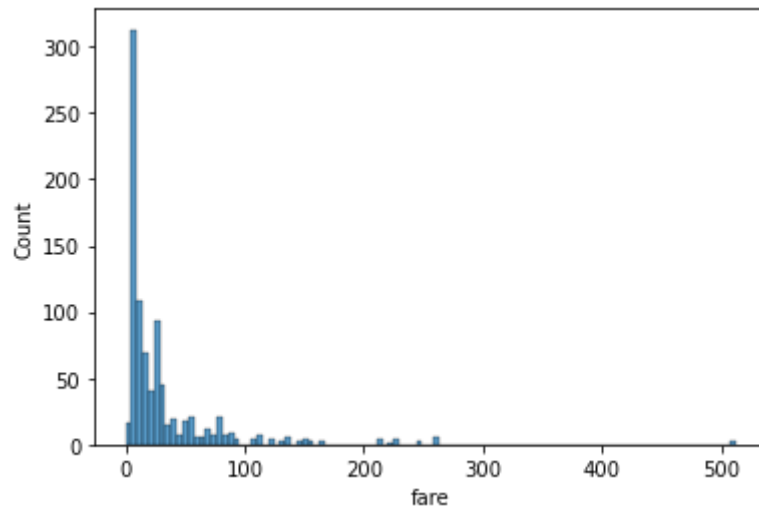
In [13]: `sns.barplot(x='sex',y='survived',hue='class',data=df)`

Out[13]: `<AxesSubplot:xlabel='sex', ylabel='survived'>`



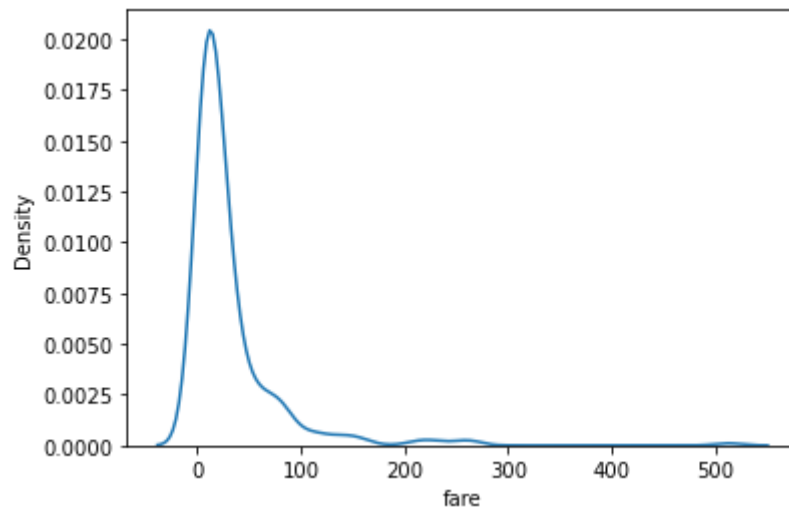
In [14]: `sns.histplot(df['fare'])`

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



In [15]: `sns.kdeplot(df['fare'])`

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



In []: