Data Visualization I

- 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.
- Write a code to check how the price of the ticket (column name: ‘fare’) for each passenger is distributed by plotting a histogram

1. 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.

In [5]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [19]:

```
data = pd.read_csv('https://raw.githubusercontent.com/dphi-official/Datasets/master/titanic_data.csv')
data.head()
```

Out[19]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [20]:

```
data.shape
```

Out[20]:

```
(891, 12)
```

In [21]:

```
data.describe()
```

Out[21]:

	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 [22]:

```
data.describe(include = 'object')
```

Out[22]:

	Name	Sex	Ticket	Cabin	Embarked
count	891	891	891	204	889
unique	891	2	681	147	3
top	Braund, Mr. Owen Harris	male	347082	B96 B98	S
freq	1	577	7	4	644

In [23]:

```
data.isnull().sum()
```

Out[23]:

```
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 [24]: data['Age'] = data['Age'].fillna(np.mean(data['Age']))
```

```
In [25]: data['Cabin'] = data['Cabin'].fillna(data['Cabin'].mode()[0])
```

```
In [31]: data['Embarked'] = data['Embarked'].fillna(data['Embarked'].mode()[0])
```

```
In [32]: data.isnull().sum()
```

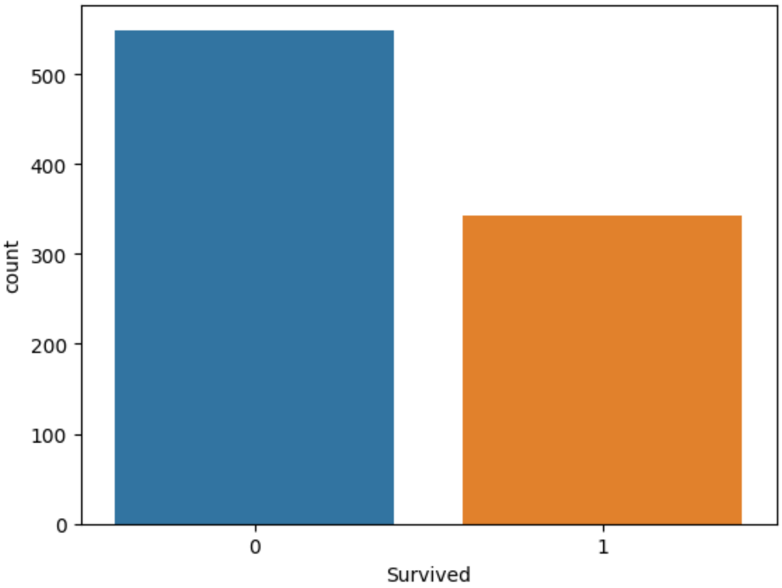
```
Out[32]: PassengerId    0
         Survived       0
         Pclass         0
         Name           0
         Sex            0
         Age            0
         SibSp          0
         Parch          0
         Ticket         0
         Fare           0
         Cabin          0
         Embarked       0
         dtype: int64
```

Countplot

- The countplot is used to represent the occurrence(counts) of the observation present in the categorical variable.

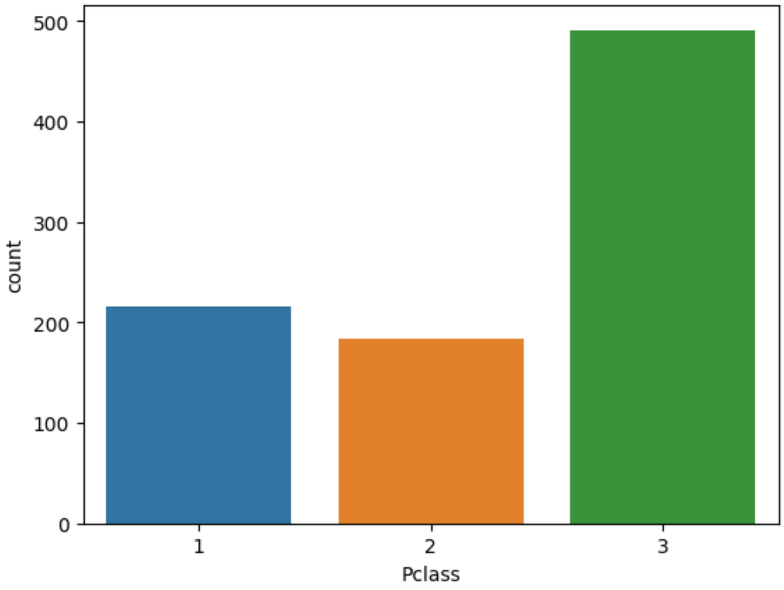
```
In [35]: sns.countplot(x='Survived',data=data)
```

```
Out[35]: <Axes: xlabel='Survived', ylabel='count'>
```



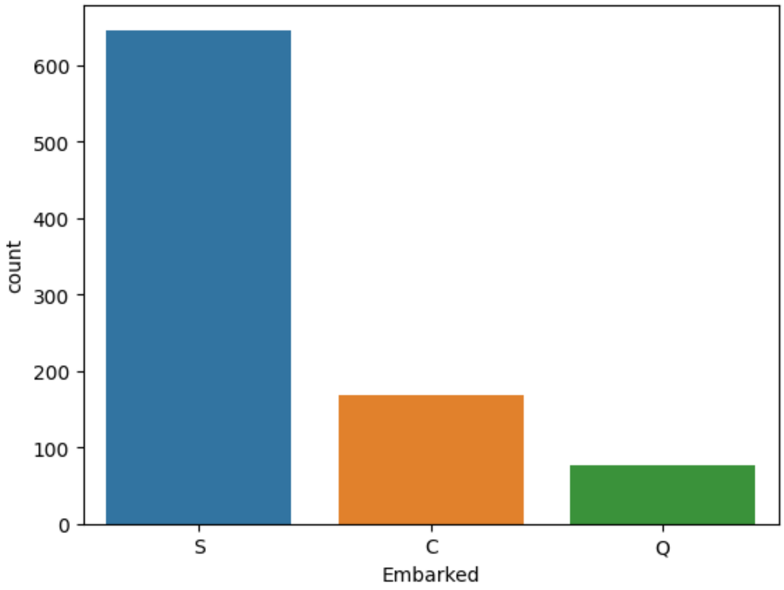
```
In [36]: sns.countplot(x='Pclass',data=data)
```

```
Out[36]: <Axes: xlabel='Pclass', ylabel='count'>
```



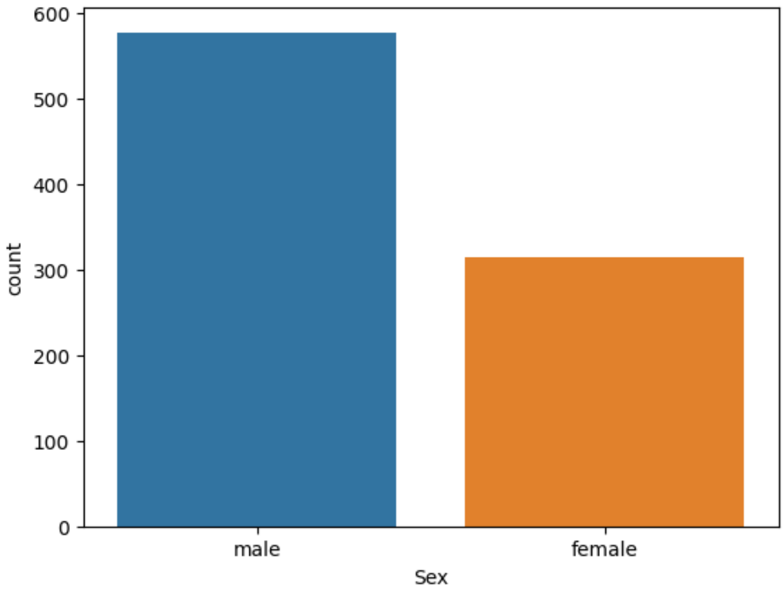
```
In [37]: sns.countplot(x='Embarked',data=data)
```

```
Out[37]: <Axes: xlabel='Embarked', ylabel='count'>
```



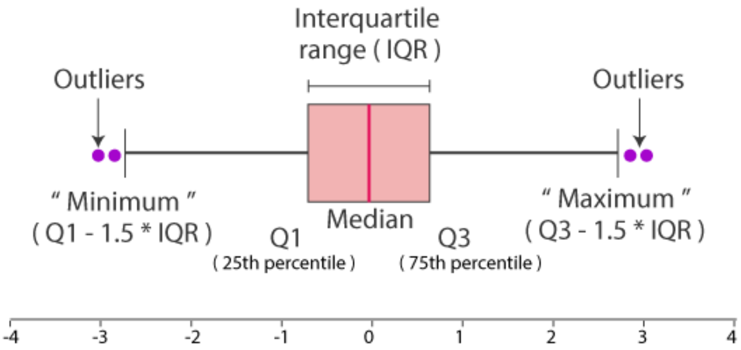
```
In [38]: sns.countplot(x='Sex',data=data)
```

```
Out[38]: <Axes: xlabel='Sex', ylabel='count'>
```



Boxplot

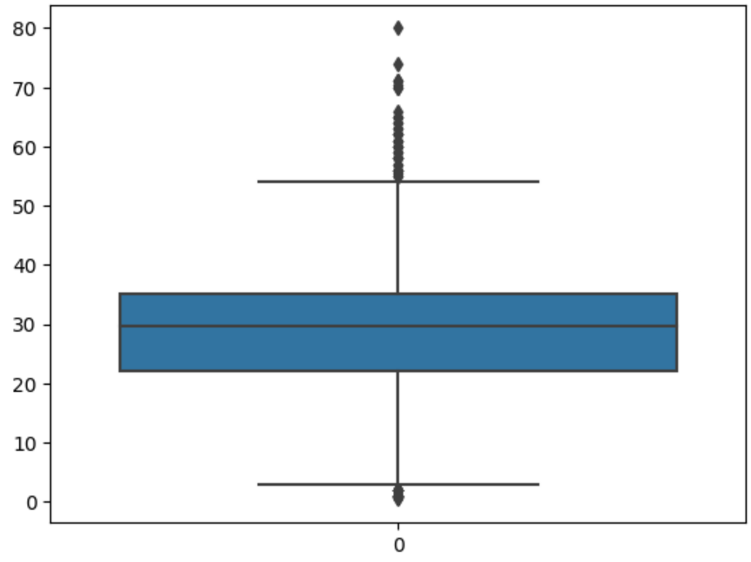
- A boxplot is a standardized way of displaying the distribution of data based on a five number summary (“minimum”, first quartile [Q1], median, third quartile [Q3] and “maximum”). It can tell you about your outliers and what their values are.



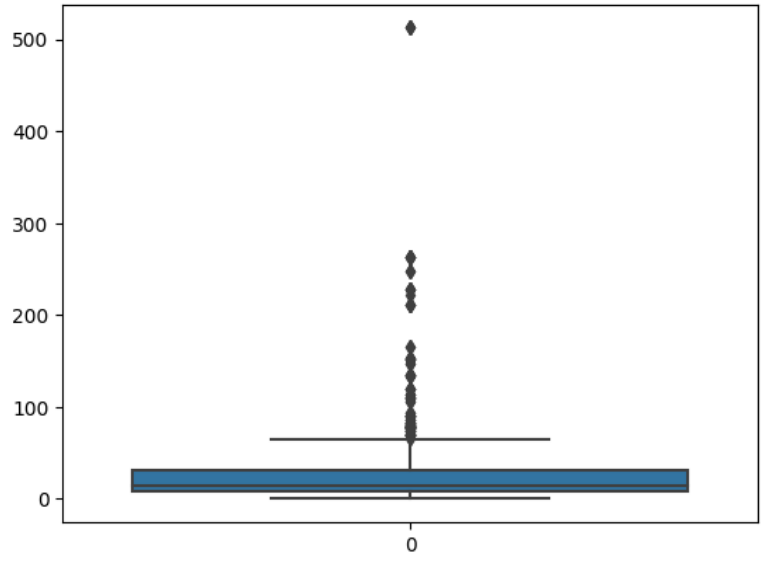
Different parts of boxplot

© Byjus.com

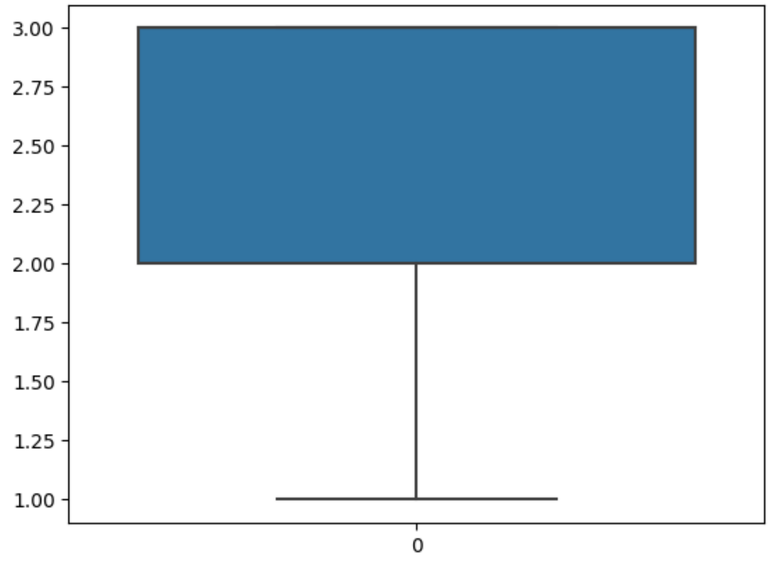
```
In [39]: sns.boxplot(data['Age'])
Out[39]: <Axes: >
```



```
In [40]: sns.boxplot(data['Fare'])
Out[40]: <Axes: >
```

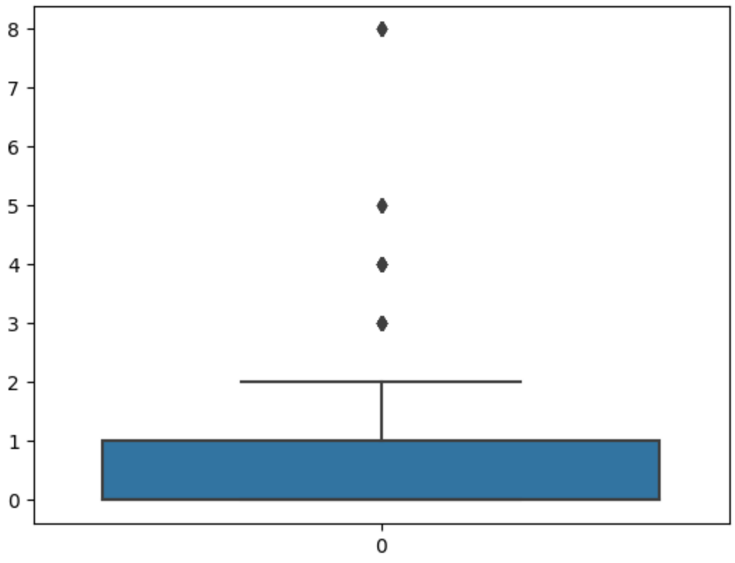


```
In [41]: sns.boxplot(data['Pclass'])
Out[41]: <Axes: >
```



```
In [42]: sns.boxplot(data['SibSp'])
```

```
Out[42]: <Axes: >
```

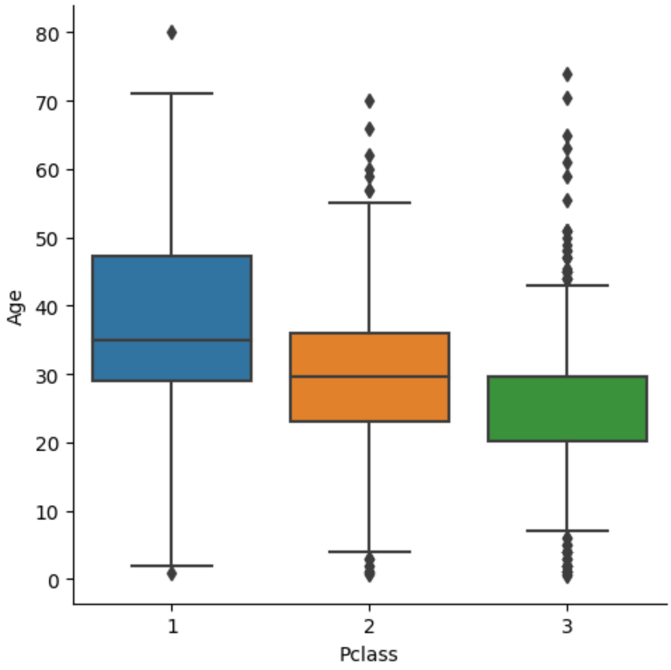


catplot

- The Seaborn catplot() function provides a figure-level interface for creating categorical plots. This means that the function allows you to map to a figure, rather than an axes object.

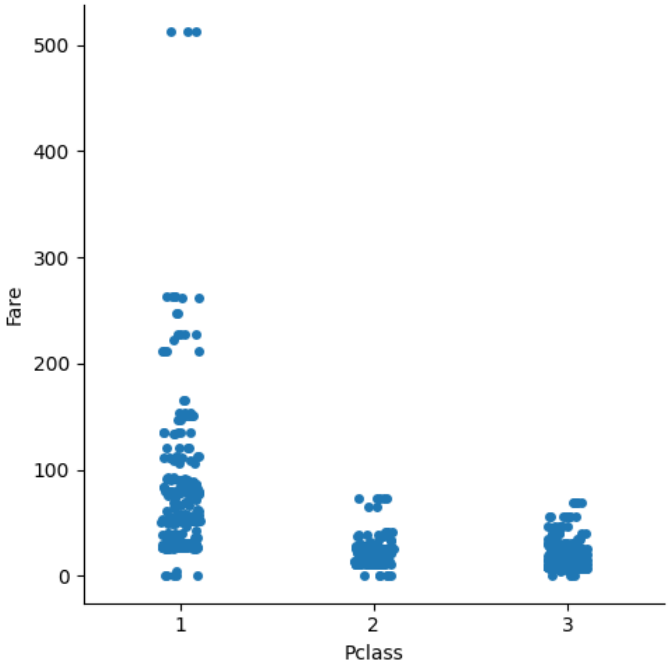
```
In [43]: sns.catplot(x= 'Pclass', y = 'Age', data=data, kind = 'box')
```

```
Out[43]: <seaborn.axisgrid.FacetGrid at 0x1913988db50>
```



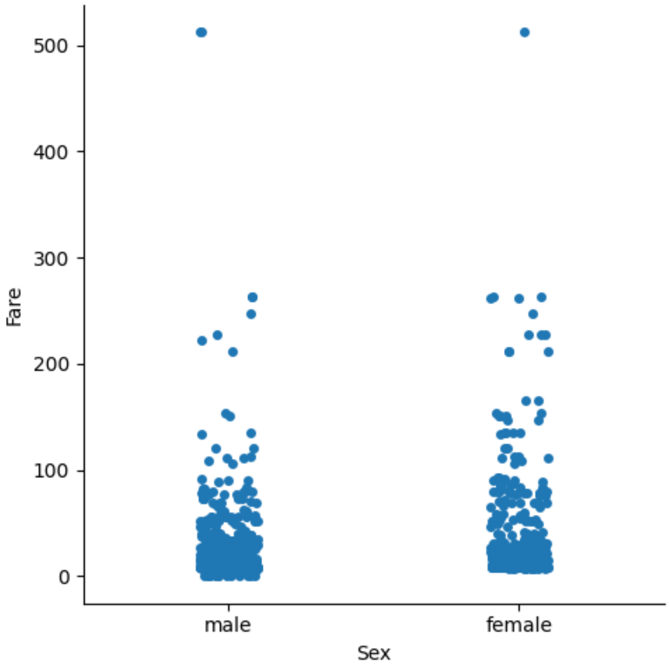
```
In [44]: sns.catplot(x= 'Pclass', y = 'Fare', data=data, kind = 'strip')
```

```
Out[44]: <seaborn.axisgrid.FacetGrid at 0x19139676c10>
```



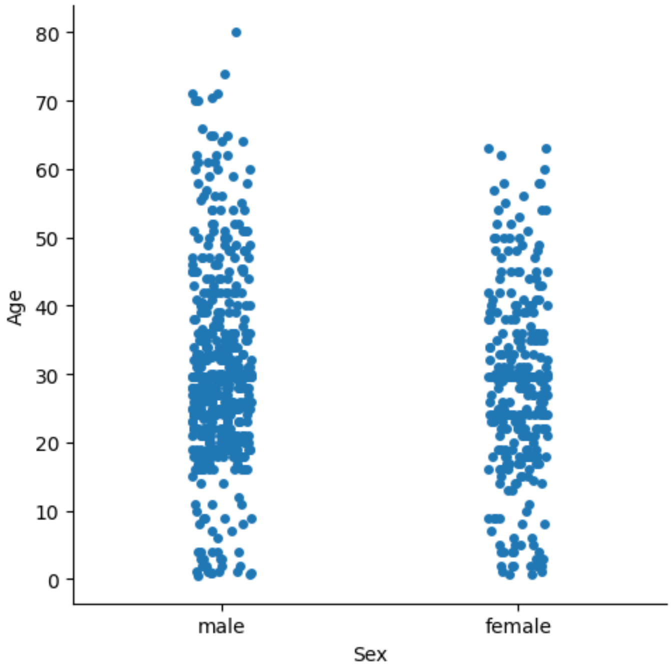
```
In [45]: sns.catplot(x= 'Sex', y = 'Fare', data=data, kind = 'strip')
```

```
Out[45]: <seaborn.axisgrid.FacetGrid at 0x19139967210>
```



```
In [46]: sns.catplot(x= 'Sex', y = 'Age', data=data, kind = 'strip')
```

```
Out[46]: <seaborn.axisgrid.FacetGrid at 0x191395fbc90>
```

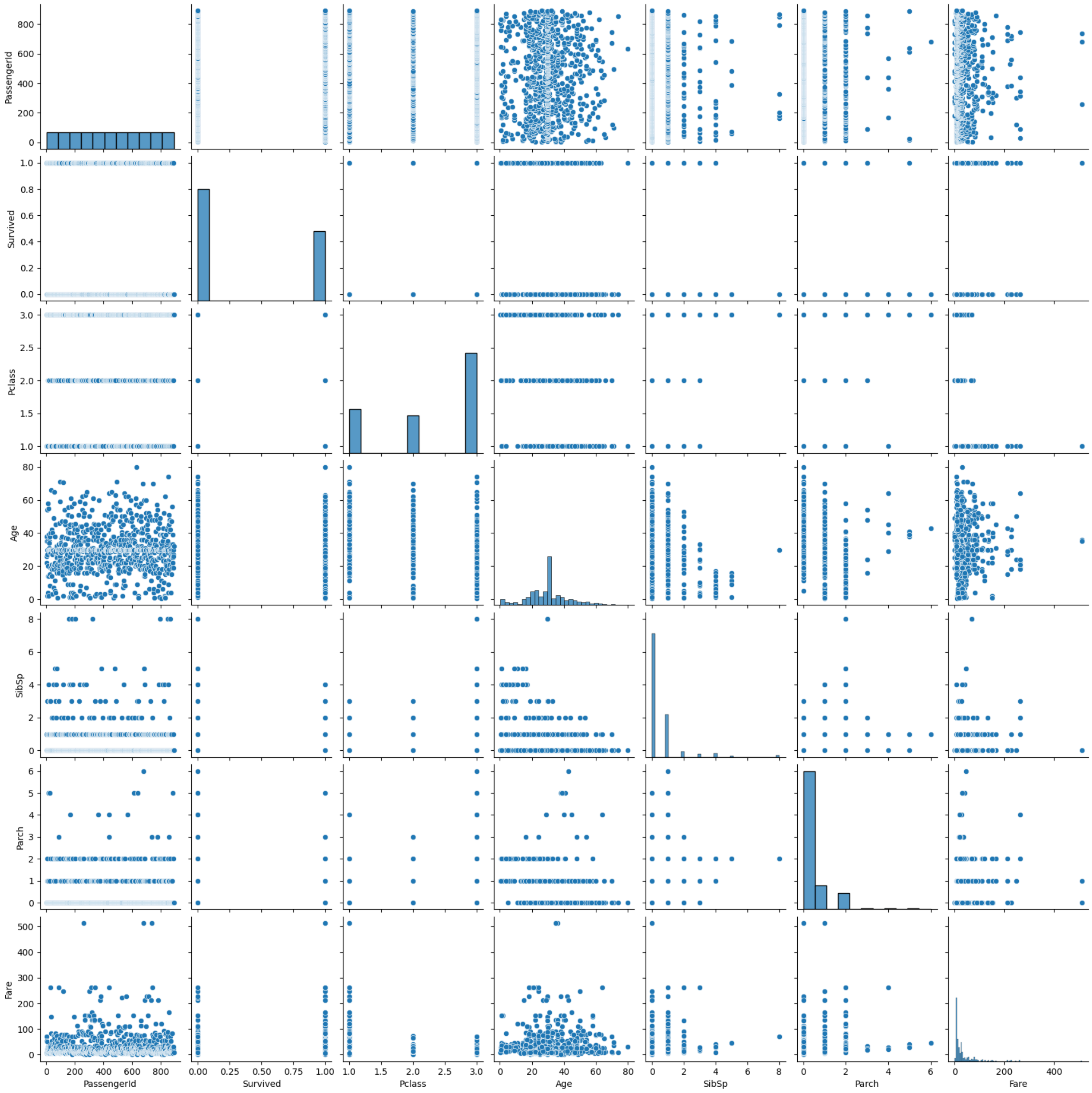


pairplot

- To plot multiple pairwise bivariate distributions in a dataset, you can use the .pairplot() function.
- The diagonal plots are the univariate plots, and this displays the relationship for the (n, 2) combination of variables in a DataFrame as a matrix of plots.

```
In [47]: sns.pairplot(data)
```

```
Out[47]: <seaborn.axisgrid.PairGrid at 0x1913aa5e010>
```



scatterplot

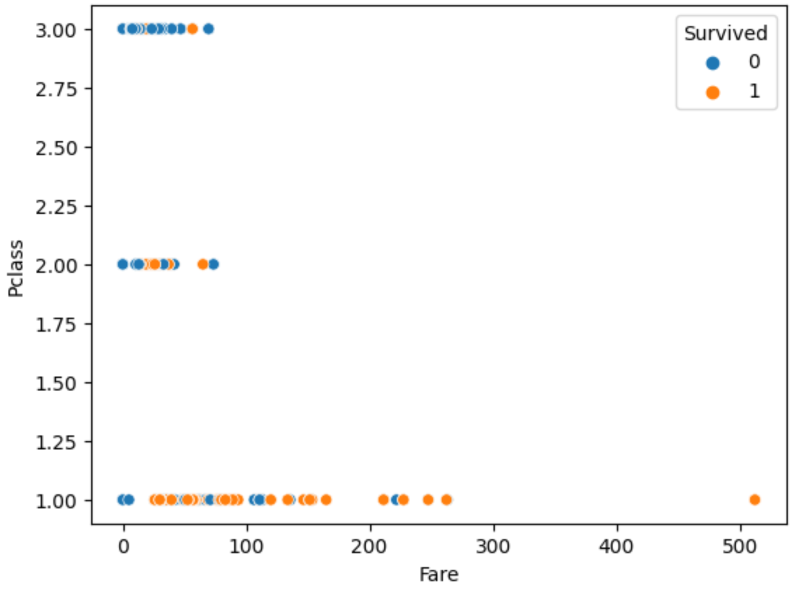
- Scatter plots are the graphs that present the relationship between two variables in a data-set. It represents data points on a two-dimensional plane or on a Cartesian system. The independent variable or attribute is plotted on the X-axis, while the dependent variable is plotted on the Y-axis. These plots are often called scatter graphs or scatter diagrams.

In [48]:

```
sns.scatterplot(x = 'Fare', y = 'Pclass', hue = 'Survived', data = data)
```

Out[48]:

```
<Axes: xlabel='Fare', ylabel='Pclass'>
```

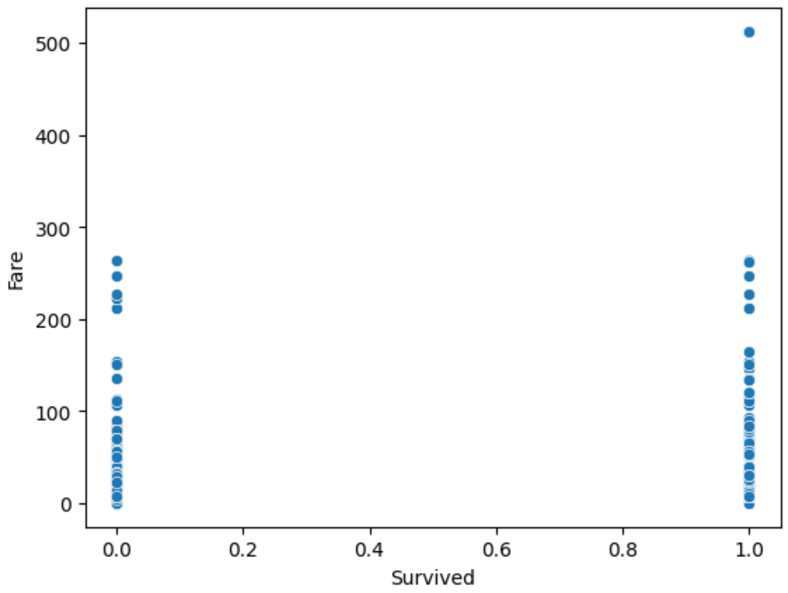


In [49]:

```
sns.scatterplot(x = 'Survived', y = 'Fare', data = data)
```

Out[49]:

```
<Axes: xlabel='Survived', ylabel='Fare'>
```

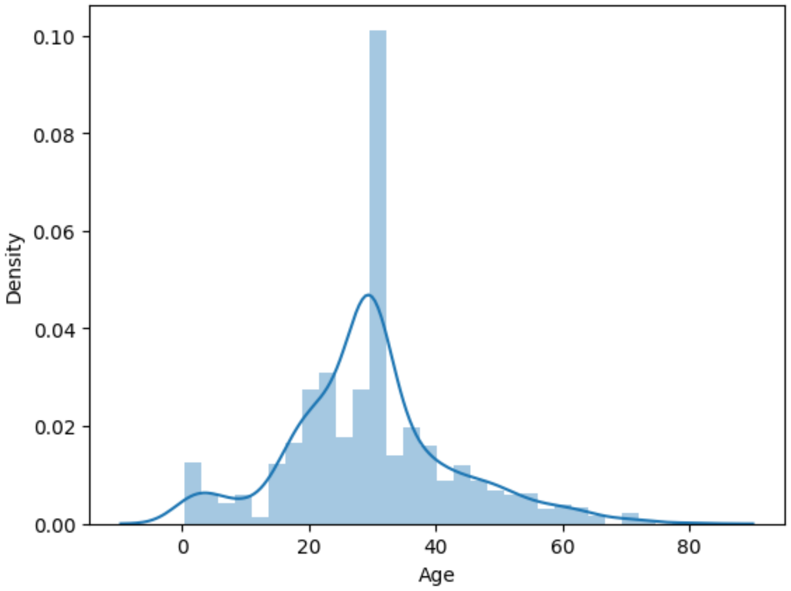


distplot

- These plots help us to visualise the distribution of data. We can use these plots to understand the mean, median, range, variance, deviation, etc of the data.

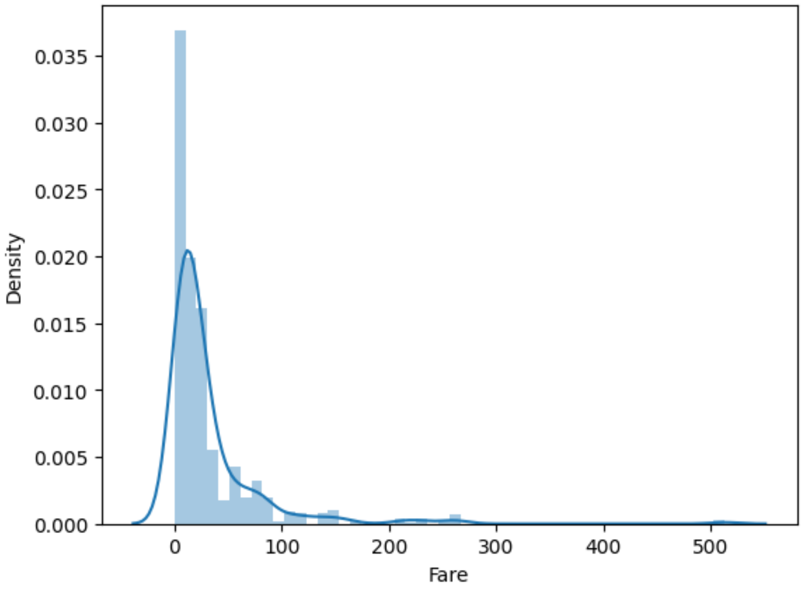
```
In [50]: sns.distplot(data['Age'])
```

```
Out[50]: <Axes: xlabel='Age', ylabel='Density'>
```



```
In [51]: sns.distplot(data['Fare'])
```

```
Out[51]: <Axes: xlabel='Fare', ylabel='Density'>
```

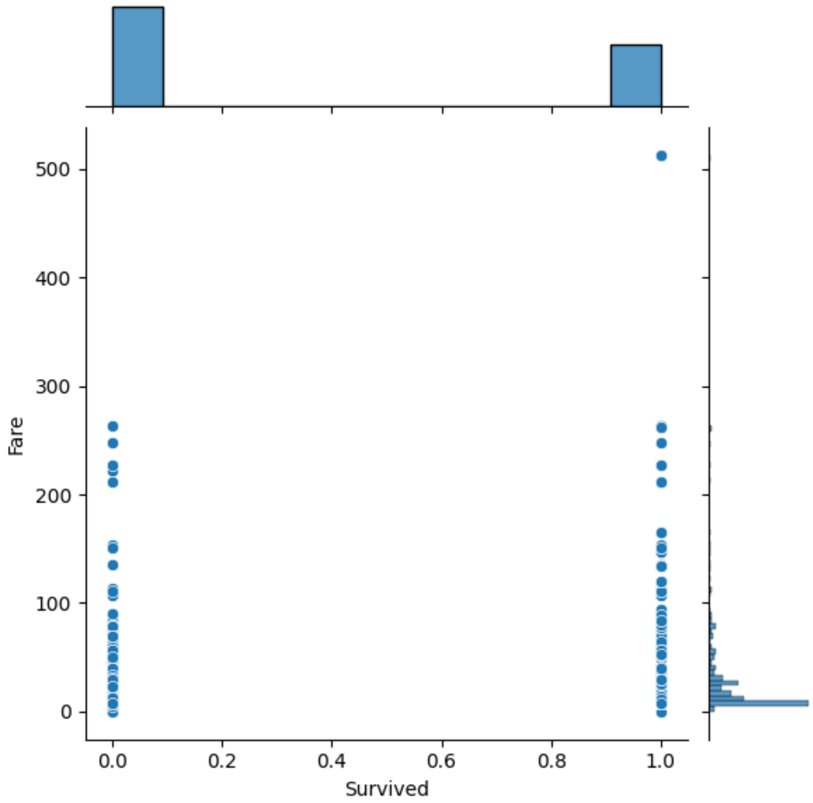


jointplot

- The joint plot is a way of understanding the relationship between two variables and the distribution of individuals of each variable.

```
In [52]: sns.jointplot(x = "Survived", y = "Fare", kind = "scatter", data = data)
```

```
Out[52]: <seaborn.axisgrid.JointGrid at 0x1913e94cad0>
```

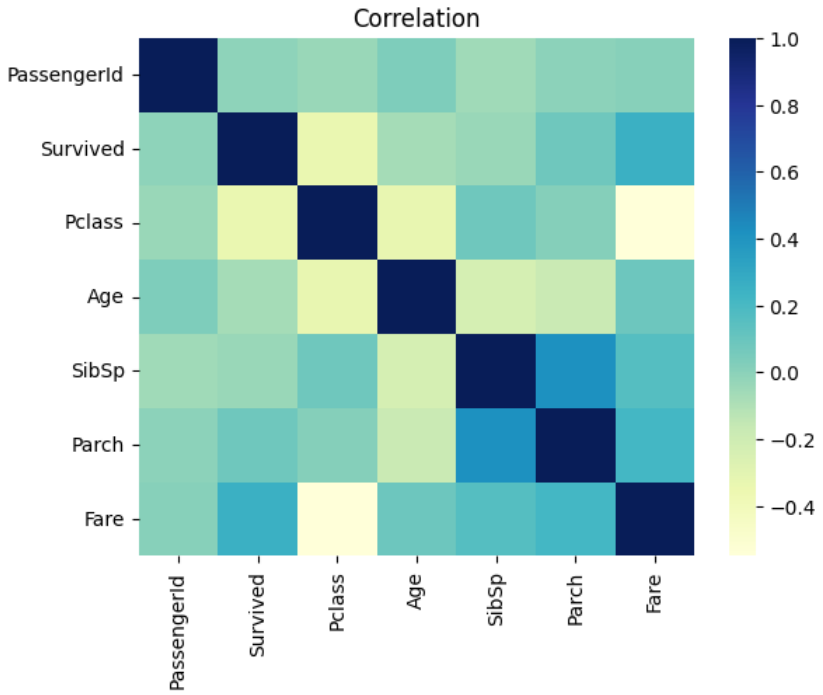


corr()

- Pandas dataframe.corr() is used to find the pairwise correlation of all columns in the Pandas Dataframe in Python.

```
In [53]: tc = data.corr()
         sns.heatmap(tc, cmap="YlGnBu")
         plt.title('Correlation')
```

```
Out[53]: Text(0.5, 1.0, 'Correlation')
```

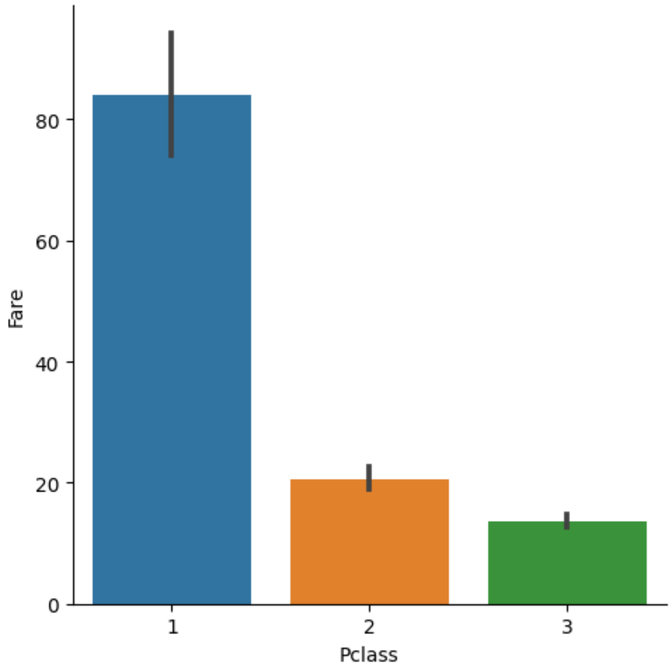


Price of Ticket for each passenger is distributed

2. Write a code to check how the price of the ticket (column name: ‘fare’) for each passenger is distributed by plotting a histogram

In [54]:
```
sns.catplot(x='Pclass', y='Fare', data=data, kind='bar')
```

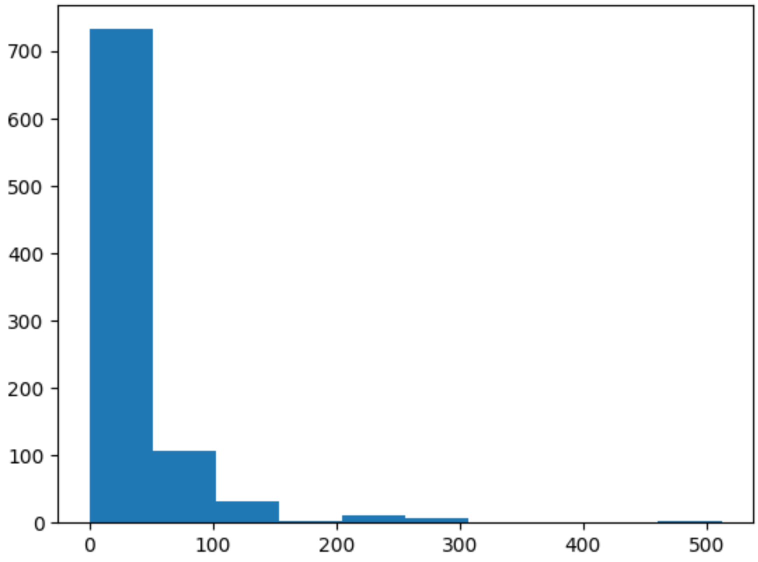
Out[54]:
```
<seaborn.axisgrid.FacetGrid at 0x1913ed76e10>
```



In [56]:
```
import matplotlib.pyplot as plt
```

In [57]:
```
plt.hist(data['Fare'])
```

Out[57]:
```
(array([732., 106.,  31.,   2.,  11.,   6.,   0.,   0.,   0.,   3.]),
 array([  0.     ,  51.23292, 102.46584, 153.69876, 204.93168, 256.1646 ,
        307.39752, 358.63044, 409.86336, 461.09628, 512.3292 ]),
 <BarContainer object of 10 artists>)
```



In [ ]: