

ASSIGNMENT 8

Data Visualization 1

1. Use the inbuilt dataset 'titanic'. Use the seaborn library to find any patterns in the data.
2. Write a code to check how the price of ticket('fare') for each passenger is distributed by plotting a histogram.

> Importing required libraries and reading the dataset

```
In [1]: ▶ import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns
```

```
In [2]: ▶ df = pd.read_csv("Titanic (1).csv")
```

In [3]:

df

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	25	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	35	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	25	0	0	STON/O2. 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	48	1	0	113803
4	5	0	3	Allen, Mr. William Henry	male	58	0	0	373450
...
886	887	0	2	Montvila, Rev. Juozas	male	12	0	0	211536
887	888	1	1	Graham, Miss. Margaret Edith	female	26	0	0	112053
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	25	1	2	W./C. 6607
889	890	1	1	Behr, Mr. Karl Howell	male	35	0	0	111369
890	891	0	3	Dooley, Mr. Patrick	male	25	0	0	370376

891 rows × 12 columns



> Data Preprocessing

```
In [4]: ▶ #checks total size(rows*columns)
df.size
```

Out[4]: 10692

```
In [5]: ▶ #checks dimensions of dataframe
df.shape
```

Out[5]: (891, 12)

```
In [6]: ▶ #checks datatype of each column
df.dtypes
```

Out[6]:

PassengerId	int64
Survived	int64
Pclass	int64
Name	object
Sex	object
Age	int64
SibSp	int64
Parch	int64
Ticket	object
Fare	float64
Cabin	object
Embarked	object
dtype:	object

```
In [7]: ▶ #checks the columns present
df.columns
```

Out[7]: Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibS
p',
 'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
 dtype='object')

```
In [8]: #checks initial statistics  
df.describe()
```

Out[8]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891
mean	446.000000	0.383838	2.308642	37.801347	0.523008	0.381594	32
std	257.353842	0.486592	0.836071	18.450415	1.102743	0.806057	49
min	1.000000	0.000000	1.000000	12.000000	0.000000	0.000000	0
25%	223.500000	0.000000	2.000000	25.000000	0.000000	0.000000	7
50%	446.000000	0.000000	3.000000	35.000000	0.000000	0.000000	14
75%	668.500000	1.000000	3.000000	45.000000	1.000000	0.000000	31
max	891.000000	1.000000	3.000000	78.000000	8.000000	6.000000	512

```
In [9]: #prints information of the dataset  
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 891 entries, 0 to 890  
Data columns (total 12 columns):  
#   Column      Non-Null Count  Dtype  
---  -  
0   PassengerId  891 non-null    int64  
1   Survived     891 non-null    int64  
2   Pclass       891 non-null    int64  
3   Name         891 non-null    object  
4   Sex          891 non-null    object  
5   Age          891 non-null    int64  
6   SibSp        891 non-null    int64  
7   Parch        891 non-null    int64  
8   Ticket       891 non-null    object  
9   Fare         891 non-null    float64  
10  Cabin        204 non-null    object  
11  Embarked     889 non-null    object  
dtypes: float64(1), int64(6), object(5)  
memory usage: 83.7+ KB
```

> Data Visualizations

Data visualization in Python is a crucial aspect of data analysis and communication. It involves creating visual representations of data to gain insights, identify patterns, and effectively convey information. Python offers several powerful libraries for data visualization, including Matplotlib, Seaborn, and Plotly.

DISTPLOTS

A distplot, short for "distribution plot," is a graphical representation that shows the distribution of a single variable or a univariate dataset. It provides a visual summary of the data's distribution, including information about the central tendency, spread, and shape of the data.

A distplot typically consists of a histogram, which divides the data into bins and displays the frequency or density of observations within each bin, along with a line or curve representing the estimated probability density function (PDF) or kernel density estimate (KDE) of the data. This helps to visualize the overall pattern and characteristics of the data distribution.

The main parameter is data, which represents the dataset to be plotted.

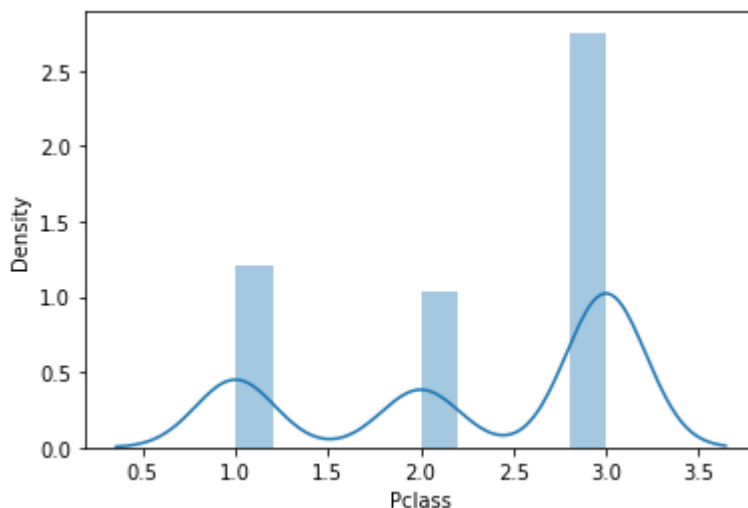
Additional optional parameters include:

- 1.bins: Specifies the number of bins to use in the histogram. Default is set automatically.
- 2.hist: Controls whether to display a histogram (True) or not (False). Default is True.
- 3.kde: Controls whether to display a kernel density estimate line (True) or not (False). Default is True.
- 4.color: Sets the color of the plot elements.
- 5.rug: Controls whether to show a rug plot of individual observations on the horizontal axis (T/F). Default is F.
- 6.label: Sets a label for the plot.
- 7.ax: Specifies the Axes object to draw the plot onto.

```
In [10]: sns.distplot(df.Pclass)
```

```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

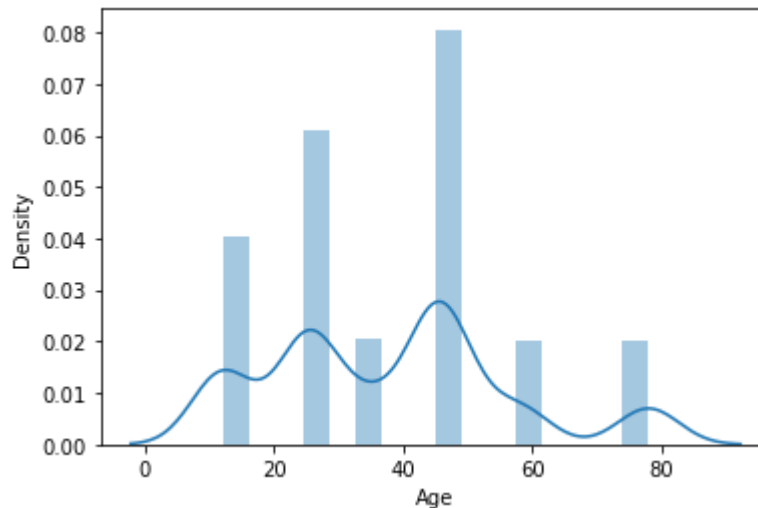
```
Out[10]: <AxesSubplot:xlabel='Pclass', ylabel='Density'>
```



```
In [11]: sns.distplot(df.Age)
```

```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

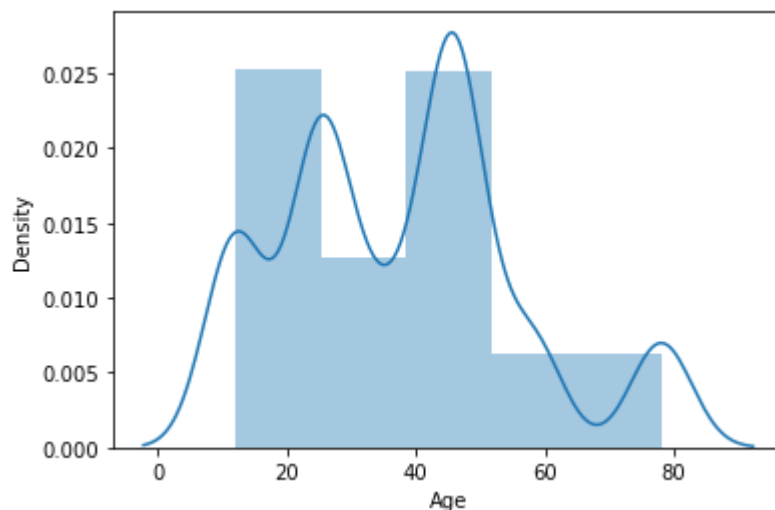
```
Out[11]: <AxesSubplot:xlabel='Age', ylabel='Density'>
```



```
In [12]: sns.distplot(df['Age'],bins=5)
```

```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
Out[12]: <AxesSubplot:xlabel='Age', ylabel='Density'>
```



```
In [13]: sns.distplot(df['Age'],bins=20,kde=False,rug=True)
```

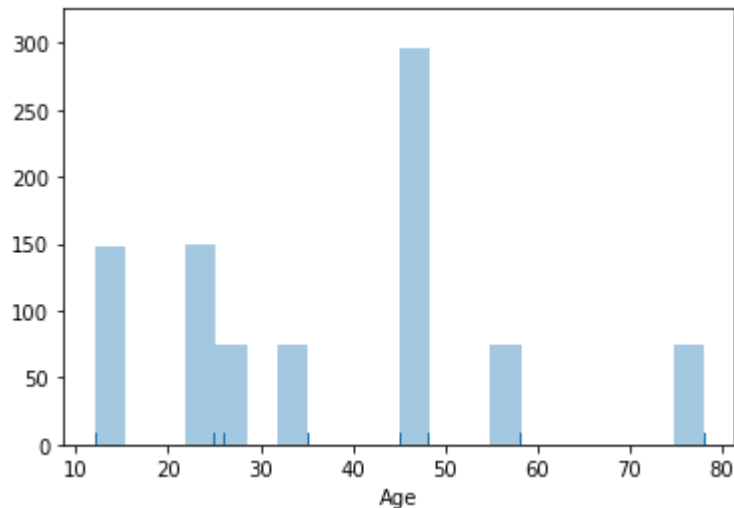
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

warnings.warn(msg, FutureWarning)

C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\distributions.py:2055: FutureWarning: The `axis` variable is no longer used and will be removed. Instead, assign variables directly to `x` or `y`.

warnings.warn(msg, FutureWarning)

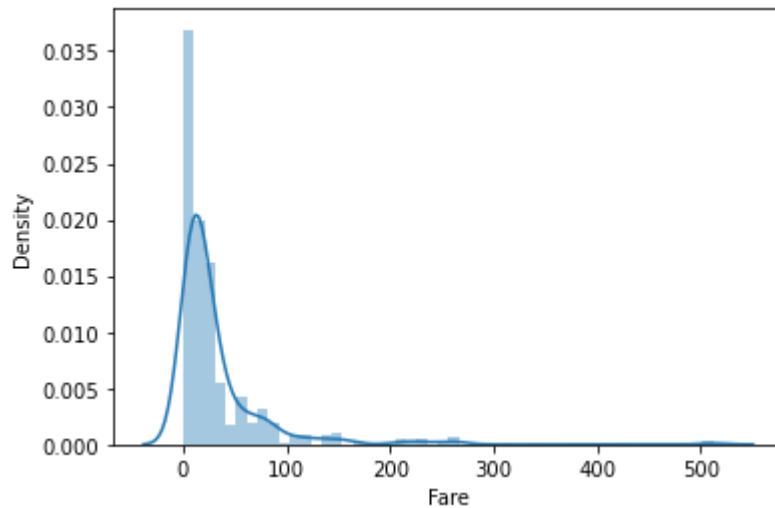
```
Out[13]: <AxesSubplot:xlabel='Age'>
```



```
In [14]: #distribution of fare plotted in a histogram  
sns.distplot(df.Fare,hist=True)
```

```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)
```

```
Out[14]: <AxesSubplot:xlabel='Fare', ylabel='Density'>
```



RUGPLOT

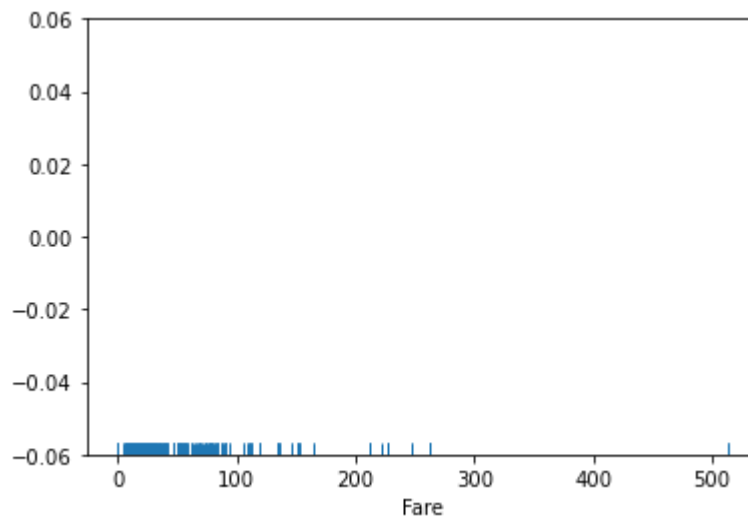
A rugplot is a simple plot in Python that displays small vertical lines or ticks along an axis to represent individual data points. It provides a visual representation of the distribution of data points along a specific axis. Rugplots are often used in combination with other plots, such as histograms or kernel density plots, to show the distribution of data points along the axis of interest.

Here are the main parameters used in Seaborn's rugplot function:

- 1.x or y: Specifies the data to be plotted along the x-axis or y-axis, respectively.
- 2.height: Sets the height of the rug ticks or lines.
- 3.axis: Specifies the axis to plot the rug ticks on. By default, axis='x' but you can set it to 'y' for the vertical axis.
- 4.palette: Sets the color palette for the rugplot.
- 5.hue: Groups the data by a categorical variable and colors the rug ticks accordingly.
- 6.orient: Determines the orientation of the rugplot. By default, it is set to 'v' for vertical orientation, but you can set it to 'h' for horizontal orientation.


```
In [15]: sns.rugplot(df['Fare'])
```

```
Out[15]: <AxesSubplot:xlabel='Fare'>
```



COUNTPLOT

A countplot is a plot in Seaborn that shows the count of observations or occurrences of each category in a categorical variable. It provides a visual representation of the distribution of categorical data.

Here are the main parameters used in Seaborn's countplot function:

- 1.data: Specifies the dataset containing the categorical variable.
- 2.x or y: Sets the variable to be plotted on the x-axis or y-axis, respectively.
- 3.hue: Groups the data by a categorical variable and colors the bars accordingly.
- 4.order: Specifies the order in which the categories should be displayed on the plot.
- 5.palette: Sets the color palette for the bars.
- 6.orient: Determines the orientation of the countplot. By default, it is set to 'v' for vertical orientation, but you can set it to 'h' for horizontal orientation.

These parameters allow you to customize the appearance and behavior of the countplot to visualize and analyze categorical data effectively.

Example:

```
import seaborn as sns
```

```
--Creating a countplot
```

```
sns.countplot(data=data, x='category', palette='Set1')
```

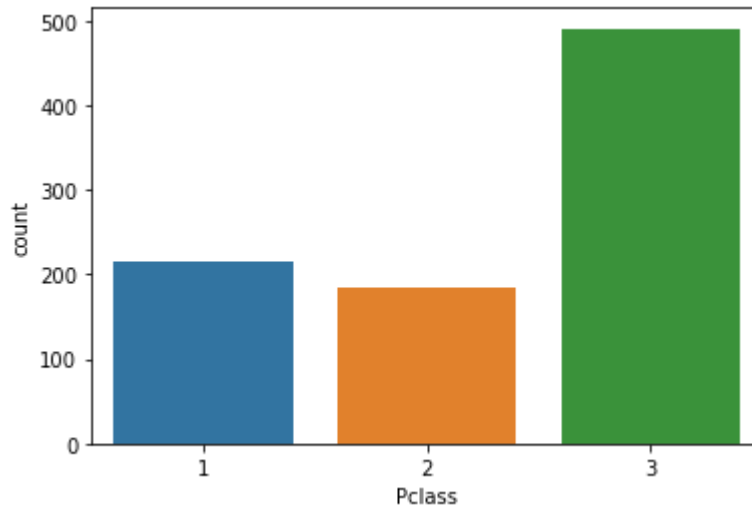
```
--Additional optional parameters
```

```
sns.countplot(data=data, y='category', hue='group', order=['A', 'B', 'C'], palette='husl', orient='h')
```

```
In [16]: sns.countplot(df.Pclass)
```

```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x.
From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
Out[16]: <AxesSubplot:xlabel='Pclass', ylabel='count'>
```



PAIRPLOT

A pairplot is a plot in Seaborn that allows us to visualize pairwise relationships between multiple variables in a dataset. It creates scatterplots for each combination of variables and histograms along the diagonal to represent the distribution of each variable.

Here are the main parameters used in Seaborn's pairplot function:

- 1.data: Specifies the dataset containing the variables to be plotted.
- 2.vars: Sets the variables to be included in the pairplot. By default, it includes all numerical variables in the dataset.
- 3.hue: Groups the data by a categorical variable and colors the scatterplots accordingly.
- 4.palette: Sets the color palette for the scatterplots.
- 5.markers: Sets the marker styles for different levels of the categorical variable.
- 6.diag_kind: Determines the type of plot to be shown on the diagonal. By default, it is set to 'auto', which shows a histogram or KDE plot based on the variable type.
- 7.plot_kws and diag_kws: Additional keyword arguments to customize the appearance of the scatterplots and diagonal plots, respectively.

Example:

```
import seaborn as sns
```

---Creating a pairplot

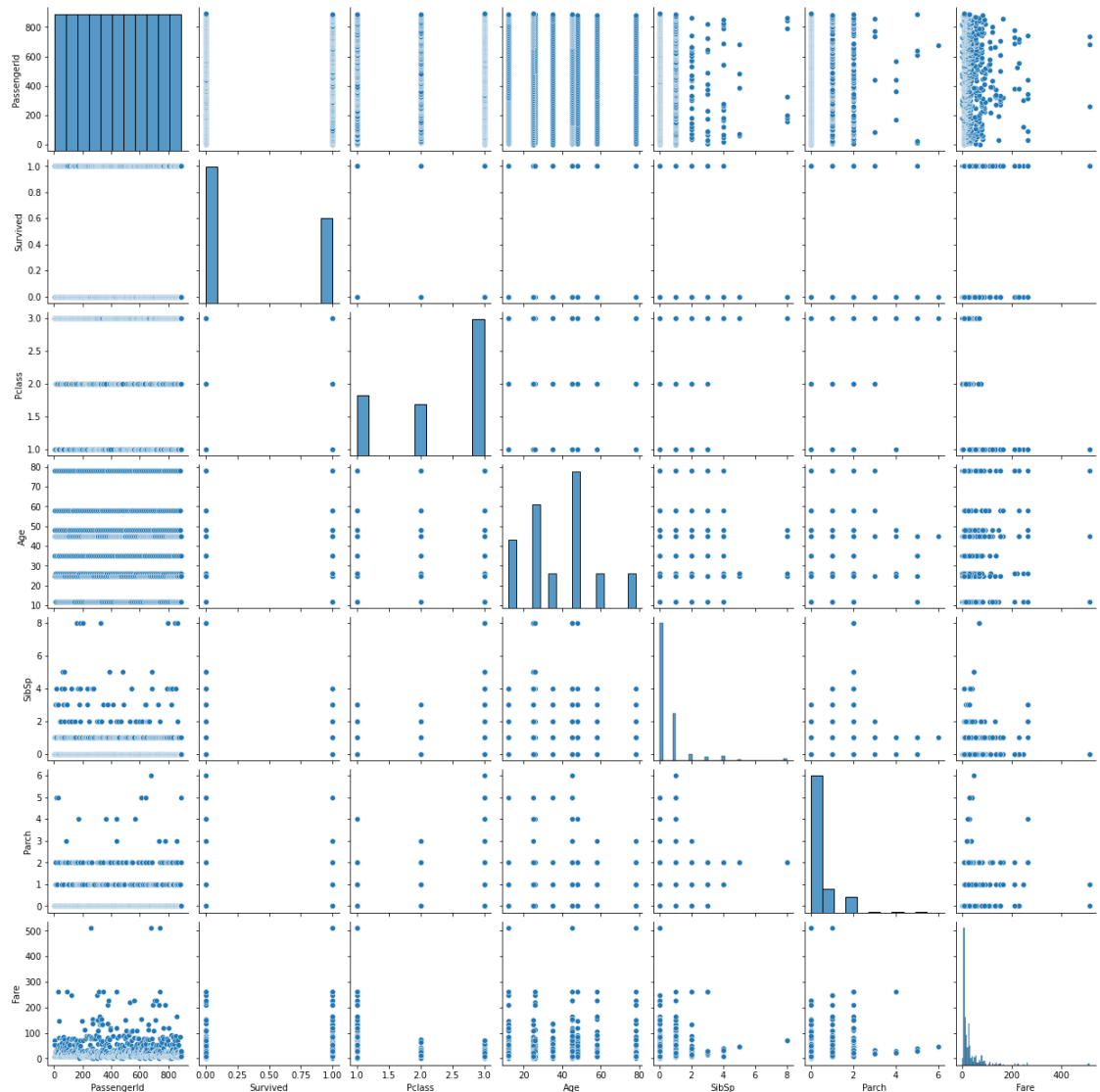
```
sns.pairplot(data=data, vars=['variable1', 'variable2', 'variable3'], hue='category', palette='Set1')
```

---Additional optional parameters

```
sns.pairplot(data=data, vars=['variable1', 'variable2', 'variable3'], hue='category', palette='Set2',  
markers=['o', 's'], diag_kind='kde', plot_kws={'alpha': 0.5}, diag_kws={'shade': True})
```

```
In [17]: ▶ sns.pairplot(df)
```

```
Out[17]: <seaborn.axisgrid.PairGrid at 0x230ce2c5310>
```



BARPLOT

A barplot is a plot in Seaborn that represents an estimate of central tendency (usually the mean) and confidence interval of a numerical variable for different levels of a categorical variable. It is useful for comparing the values of a variable across different categories.

Here are the main parameters used in Seaborn's barplot function:

- 1.data: Specifies the dataset containing the variables to be plotted.
- 2.x or y: Sets the categorical variable to be plotted on the x-axis or y-axis, respectively.
- 3.hue: Groups the data by a categorical variable and colors the bars accordingly.
- 4.estimator: Determines the statistical function to estimate the central tendency of the numerical variable within each category. By default, it computes the mean.
- 5.ci: Sets the size of the confidence interval for the central tendency estimate. By default, it is set to 95%.
- 6.palette: Sets the color palette for the bars.
- 7.order: Specifies the order in which the categories should be displayed on the plot.
- 8.orient: Determines the orientation of the barplot. By default, it is set to 'v' for vertical orientation, but you can set it to 'h' for horizontal orientation.

Example:

```
import seaborn as sns
```

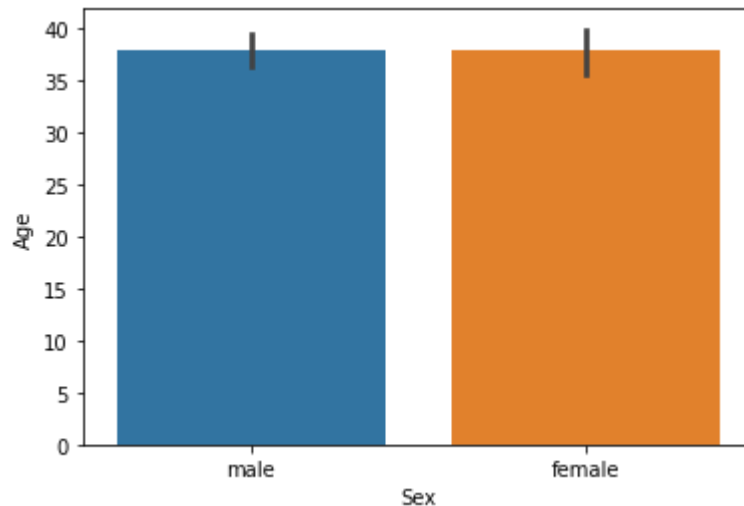
```
---Creating a barplot sns.barplot(data=data, x='category', y='value', hue='group', palette='Set1')
```

```
---Additional optional parameters sns.barplot(data=data, x='category', y='value', hue='group',  
estimator=np.median, ci=90, palette='Set2', order=['A', 'B', 'C'], orient='h')
```

```
In [18]: sns.barplot(df.Sex,df.Age)
```

C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in a n error or misinterpretation.
warnings.warn(

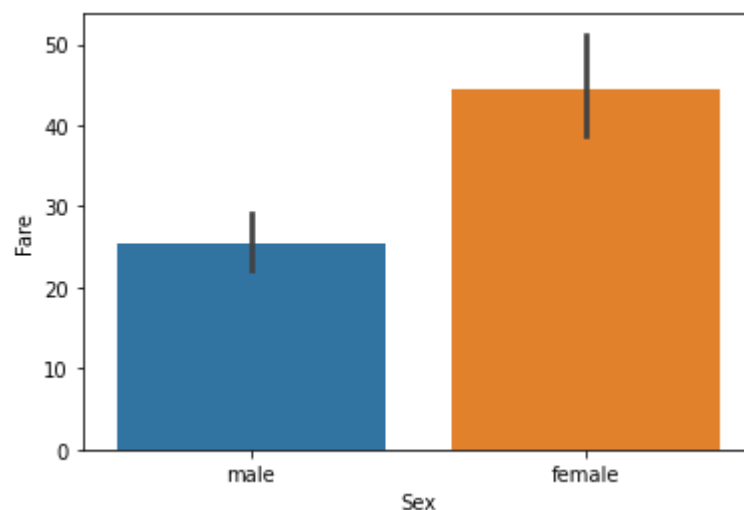
```
Out[18]: <AxesSubplot:xlabel='Sex', ylabel='Age'>
```



```
In [19]: sns.barplot(df.Sex,df.Fare)
```

C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in a n error or misinterpretation.
warnings.warn(

```
Out[19]: <AxesSubplot:xlabel='Sex', ylabel='Fare'>
```



JOINTPLOT

A jointplot is a plot in Seaborn that combines two different types of plots to visualize the relationship between two variables: a scatterplot and a histogram or kernel density estimate (KDE) plot.

Here are the main parameters used in Seaborn's jointplot function:

- 1.data: Specifies the dataset containing the variables to be plotted.
- 2.x and y: Sets the variables to be plotted on the x-axis and y-axis, respectively.
- 3.kind: Determines the type of plot to be shown. It can be set to 'scatter' for a scatterplot, 'reg' for a scatterplot with a linear regression line, 'resid' for a scatterplot of the residuals, 'kde' for a 2D kernel density estimate plot, or 'hex' for a hexbin plot.
- 4.hue: Groups the data by a categorical variable and colors the data points or KDE contours accordingly.
- 5.height: Sets the height of the figure in inches.
- 6.ratio: Sets the ratio of joint axes size to marginal axes size.
- 7.space: Sets the space between the joint and marginal axes.
- 8.dropna: Controls whether to drop missing values or not.
- 9.color: Sets the color of the plots.
- 10.xlim and ylim: Sets the limits of the x-axis and y-axis, respectively.

Example:

```
import seaborn as sns
```

```
--Creating a jointplot sns.jointplot(data=data, x='x_variable', y='y_variable', kind='scatter',  
hue='category', height=6, ratio=5, space=0.2, xlim=(-10, 10), ylim=(-5, 5), color='blue')
```

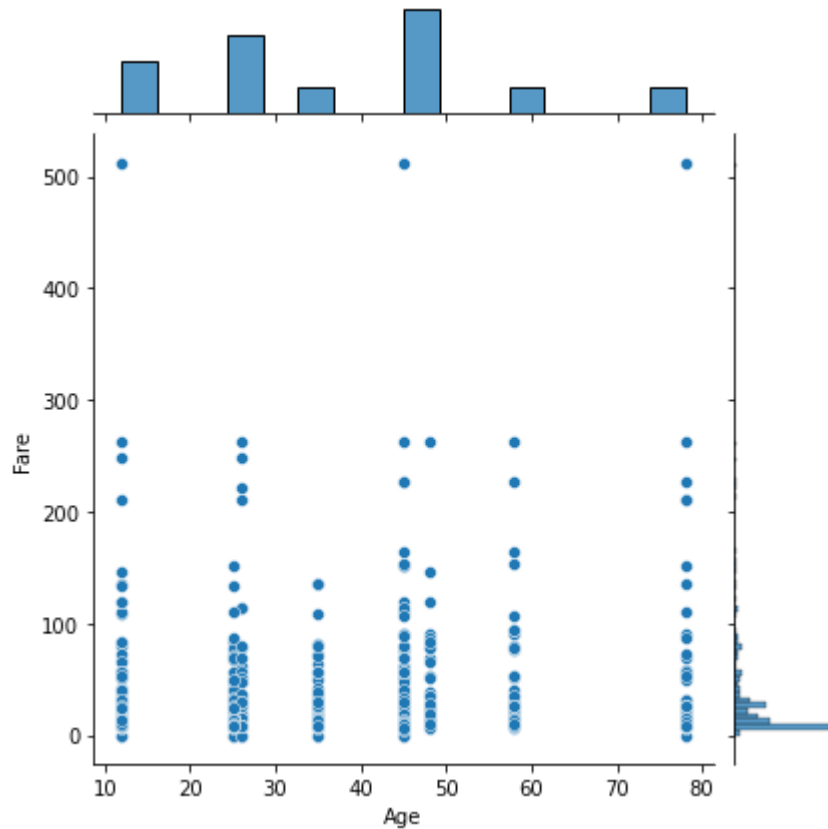
```
--Additional optional parameters sns.jointplot(data=data, x='x_variable', y='y_variable',  
kind='kde', hue='category', color='green')
```

```
In [20]: sns.jointplot(df.Age,df.Fare,kind='scatter')
```

C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in a n error or misinterpretation.

```
warnings.warn(
```

```
Out[20]: <seaborn.axisgrid.JointGrid at 0x230d0a78dc0>
```

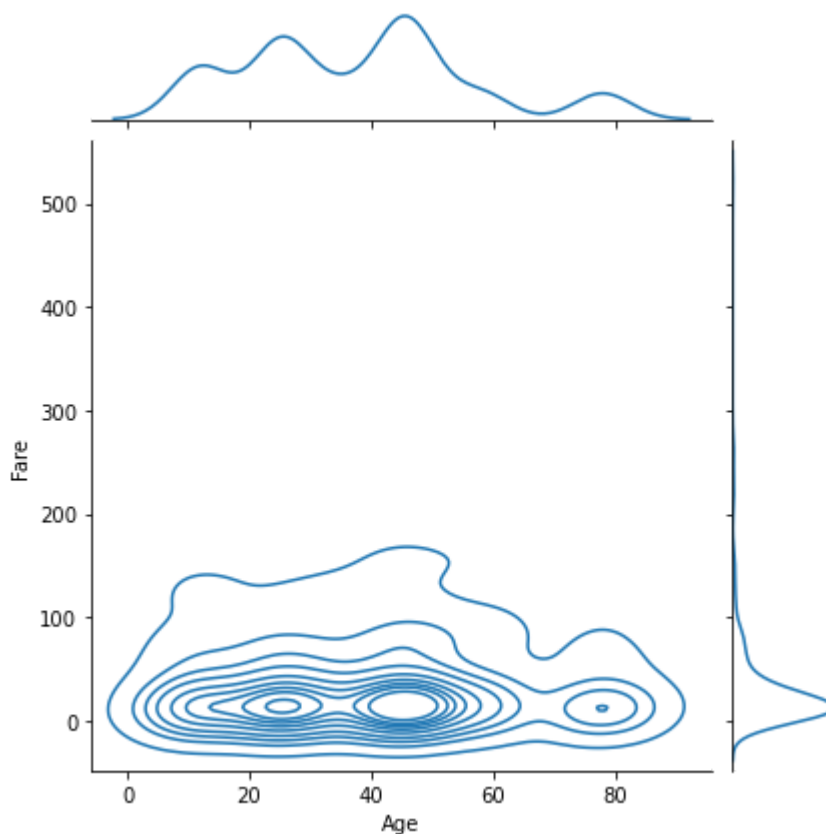


```
In [21]: sns.jointplot(df.Age,df.Fare,kind='kde')
```

C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in a n error or misinterpretation.

```
warnings.warn(
```

```
Out[21]: <seaborn.axisgrid.JointGrid at 0x230d0969040>
```

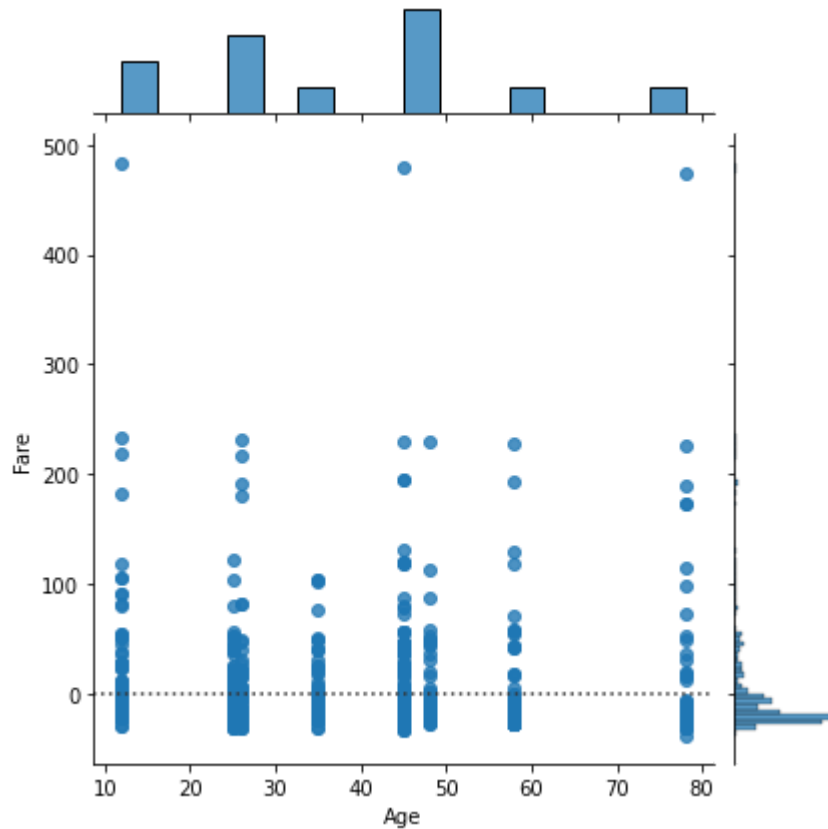



```
In [22]: sns.jointplot(df.Age,df.Fare,kind='resid')
```

C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in a n error or misinterpretation.

```
warnings.warn(
```

```
Out[22]: <seaborn.axisgrid.JointGrid at 0x230d14e4a00>
```

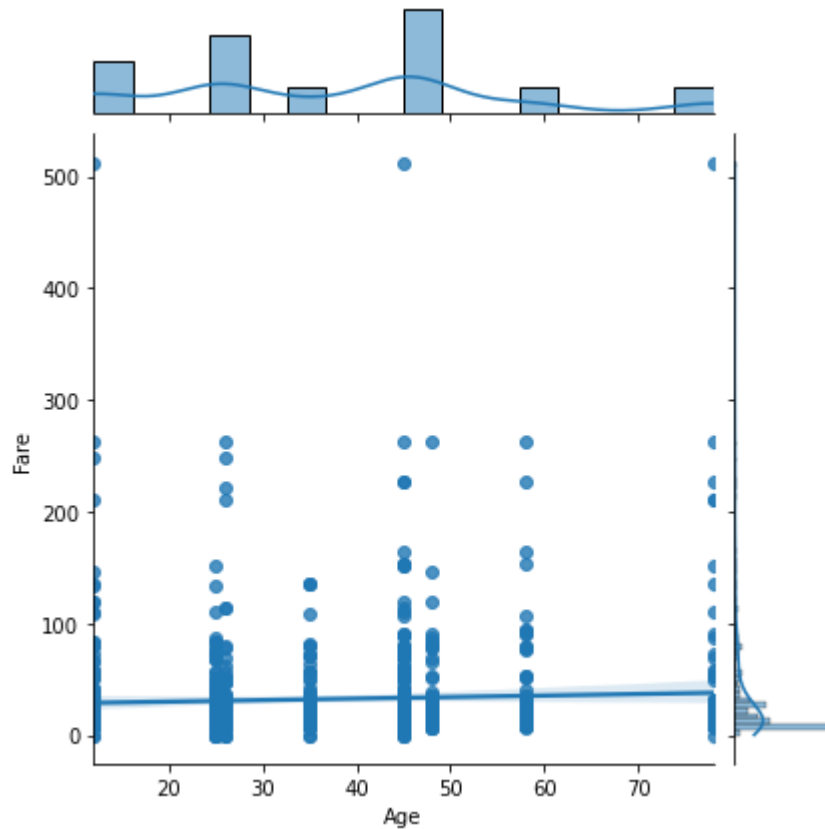


```
In [23]: sns.jointplot(df.Age,df.Fare,kind='reg')
```

C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in a n error or misinterpretation.

```
warnings.warn(
```

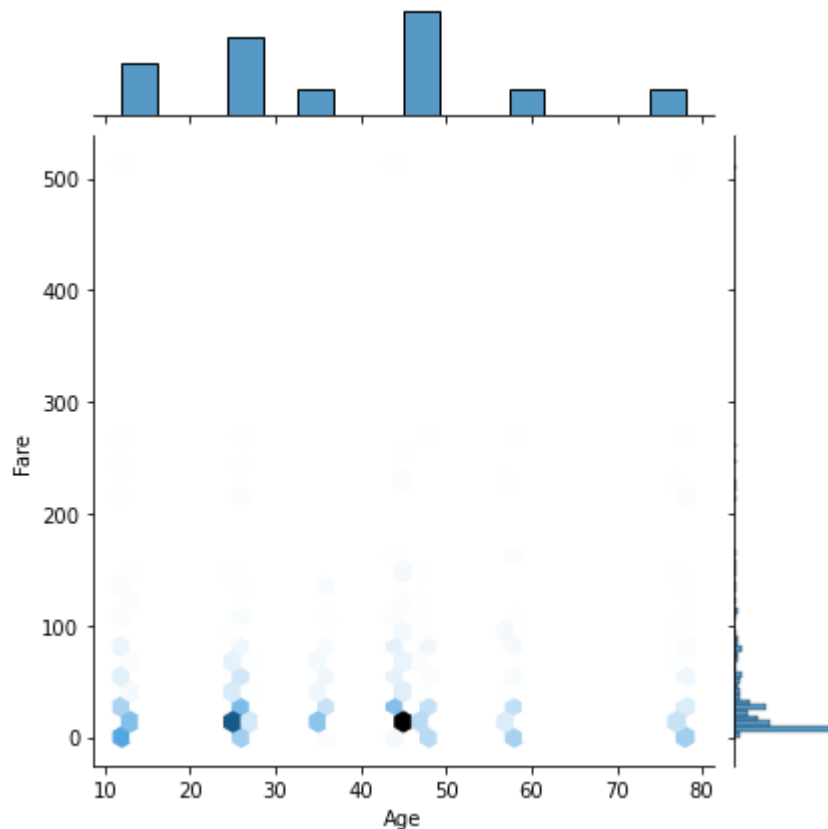
```
Out[23]: <seaborn.axisgrid.JointGrid at 0x230d176f190>
```



```
In [24]: sns.jointplot(df.Age,df.Fare,kind='hex')
```

```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in a n error or misinterpretation.
  warnings.warn(
```

```
Out[24]: <seaborn.axisgrid.JointGrid at 0x230d1879fd0>
```



SWARMPLOT

A swarmplot is a plot in Seaborn that displays the distribution of categorical data points along a numerical axis. It positions each data point horizontally, avoiding overlapping, to provide a visual representation of the data density.

Here are the main parameters used in Seaborn's swarmplot function:

- 1.data: Specifies the dataset containing the variables to be plotted.
- 2.x or y: Sets the categorical variable to be plotted on the x-axis or y-axis, respectively.
- 3.hue: Groups the data by a categorical variable and colors the data points accordingly.
- 4.order: Specifies the order in which the categories should be displayed on the plot.
- 5.palette: Sets the color palette for the data points.
- 6.size: Sets the size of the data points.
- 7.edgecolor: Sets the color of the edges of the data points.
- 8.alpha: Sets the transparency level of the data points.
- 9.orient: Determines the orientation of the swarmplot. By default, it is set to 'v' for vertical orientation, but you can set it to 'h' for horizontal orientation.

```
In [25]: sns.swarmplot(df['Sex'],df['Fare'])
```

```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
```

```
warnings.warn(
```

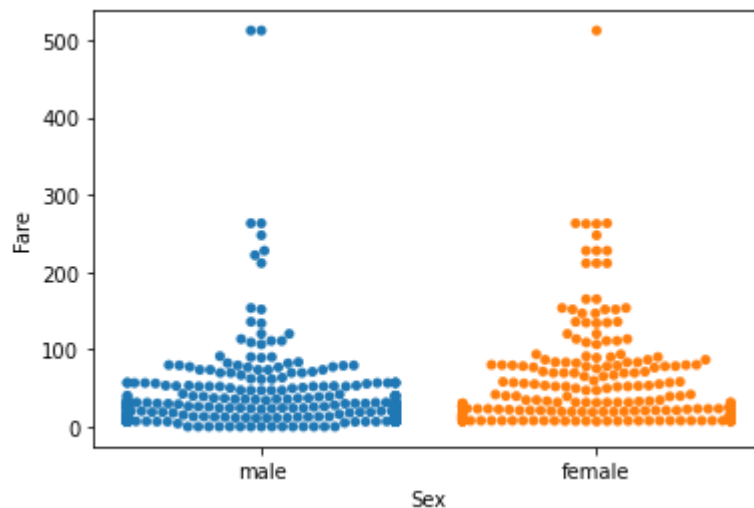
```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 71.1% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
```

```
warnings.warn(msg, UserWarning)
```

```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 48.4% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
```

```
warnings.warn(msg, UserWarning)
```

```
Out[25]: <AxesSubplot:xlabel='Sex', ylabel='Fare'>
```



```
In [26]: sns.swarmplot(df['Sex'],df['Fare'],hue=df['Survived'])
```

```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\_decorator
s.py:36: FutureWarning: Pass the following variables as keyword args: x,
y. From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in a
n error or misinterpretation.
```

```
warnings.warn(
```

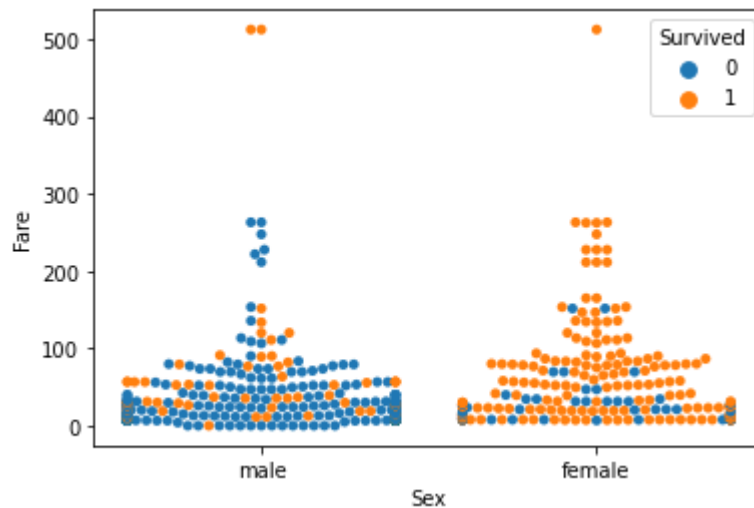
```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\categorical
l.py:1296: UserWarning: 71.1% of the points cannot be placed; you may wa
nt to decrease the size of the markers or use stripplot.
```

```
warnings.warn(msg, UserWarning)
```

```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\categorical
l.py:1296: UserWarning: 48.4% of the points cannot be placed; you may wa
nt to decrease the size of the markers or use stripplot.
```

```
warnings.warn(msg, UserWarning)
```

```
Out[26]: <AxesSubplot:xlabel='Sex', ylabel='Fare'>
```



SCATTER PLOT

A scatterplot is a plot in Seaborn that displays the relationship between two numerical variables as a collection of data points. It is useful for visualizing the distribution, pattern, and correlation between two variables.

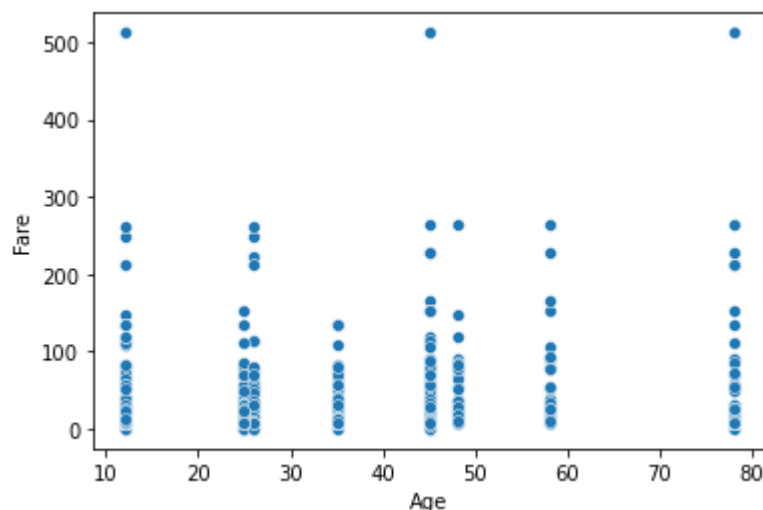
Here are the main parameters used in Seaborn's scatterplot function:

- 1.data: Specifies the dataset containing the variables to be plotted.
- 2.x and y: Sets the variables to be plotted on the x-axis and y-axis, respectively.
- 3.hue: Groups the data by a categorical variable and colors the data points accordingly.
- 4.style: Groups the data by a categorical variable and assigns different marker styles to the data points.
- 5.size: Sets the size of the data points.
- 6.palette: Sets the color palette for the data points.
- 7.alpha: Sets the transparency level of the data points.
- 8.x_jitter and y_jitter: Adds random noise to the x-axis and y-axis, respectively, to reduce overlapping data points.
- 9.legend: Controls whether to display the legend or not.
- 10.ax: Specifies the Axes object to draw the scatterplot onto.

In [27]: `sns.scatterplot(df.Age,df.Fare)`

```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\_decorator
s.py:36: FutureWarning: Pass the following variables as keyword args: x,
y. From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in a
n error or misinterpretation.
  warnings.warn(
```

Out[27]: `<AxesSubplot:xlabel='Age', ylabel='Fare'>`

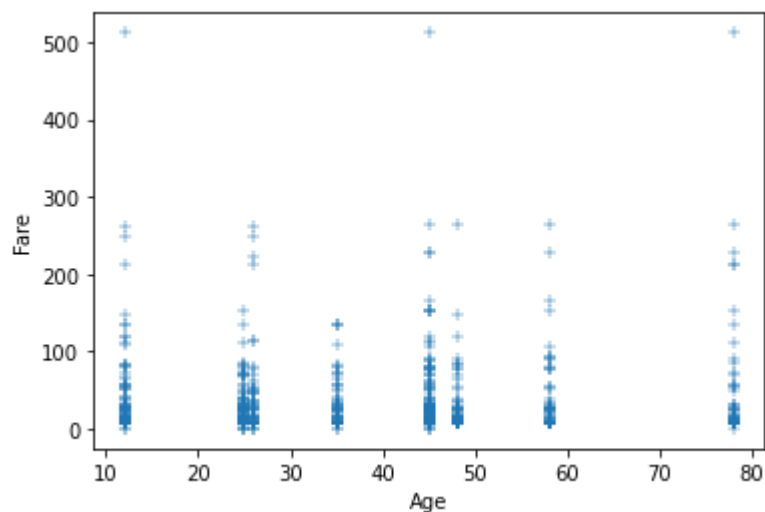


```
In [28]: sns.scatterplot(df.Age,df.Fare,marker='+')
```

C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in a n error or misinterpretation.

```
warnings.warn(
```

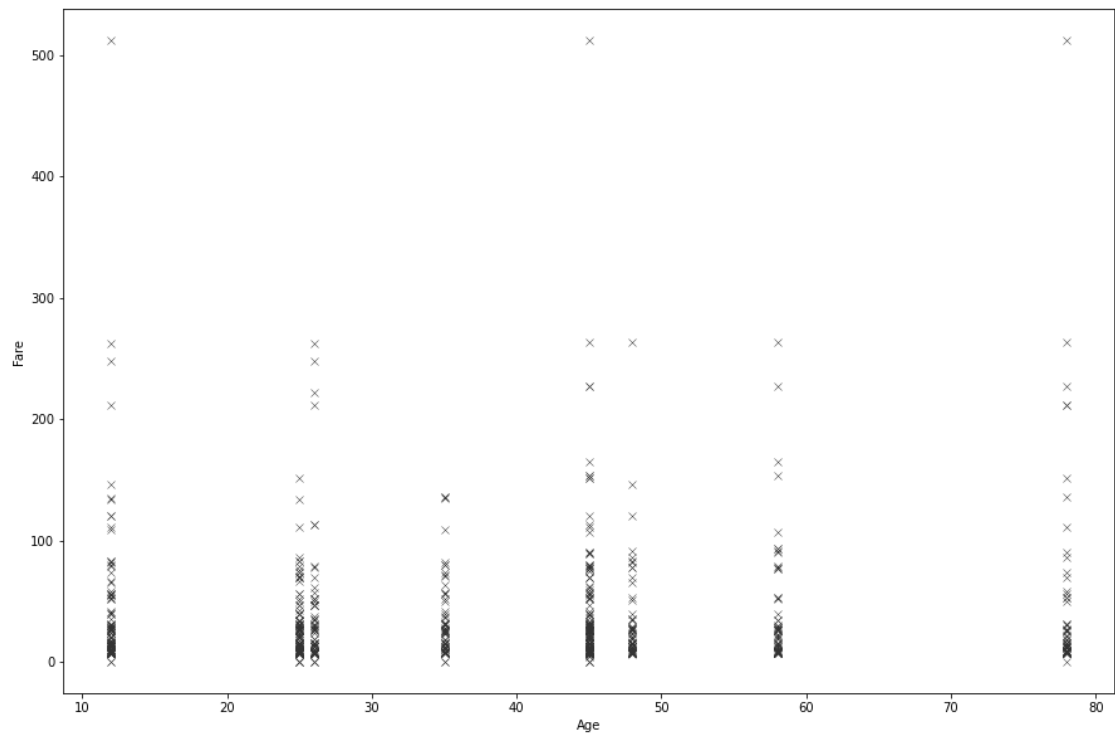
```
Out[28]: <AxesSubplot:xlabel='Age', ylabel='Fare'>
```



```
In [29]: plt.figure(figsize=(15,10))
sns.scatterplot(df.Age,df.Fare,marker='x',color=".2")
```

```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\_decorator
s.py:36: FutureWarning: Pass the following variables as keyword args: x,
y. From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in a
n error or misinterpretation.
  warnings.warn(
```

```
Out[29]: <AxesSubplot:xlabel='Age', ylabel='Fare'>
```



STRIPLOT

A stripplot is a plot in Seaborn that displays the distribution of categorical data points along a numerical axis. It positions each data point horizontally, allowing for better visibility of individual observations and their density.

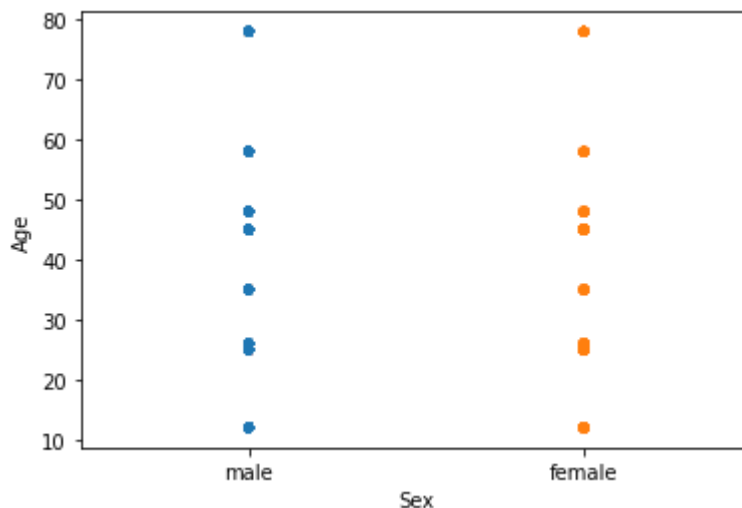
Here are the main parameters used in Seaborn's stripplot function:

- 1.data: Specifies the dataset containing the variables to be plotted.
- 2.x or y: Sets the categorical variable to be plotted on the x-axis or y-axis, respectively.
- 3.hue: Groups the data by a categorical variable and colors the data points accordingly.
- 4.order: Specifies the order in which the categories should be displayed on the plot.
- 5.palette: Sets the color palette for the data points.
- 6.size: Sets the size of the data points.
- 7.edgecolor: Sets the color of the edges of the data points.
- 8.alpha: Sets the transparency level of the data points.
- 9.jitter: Adds random noise to the data points along the categorical axis to reduce overlapping.
- 10.orient: Determines the orientation of the stripplot. By default, it is 'vertical'.

```
In [30]: sns.stripplot(df.Sex,df.Age,jitter=False)
```

```
C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
Out[30]: <AxesSubplot:xlabel='Sex', ylabel='Age'>
```



```
In [31]: sns.stripplot(df.Sex,df.Age,jitter=True,hue=df.Survived)
```

C:\Users\Shravani Sajekar\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in a n error or misinterpretation.

```
warnings.warn(
```

```
Out[31]: <AxesSubplot:xlabel='Sex', ylabel='Age'>
```

