import seaborn as sns

from matplotlib import pyplot as plt

\_

df=sns.load\_dataset('titanic')

df.head()

\_

plt.figure(figsize=(6,4))

sns.displot(df['age']) # Use sns.distplot(df['age']) for older versions of seaborn library

plt.show()

\_

plt.figure(figsize=(5,3))

bp = sns.boxplot(x='class',y='age',palette='pastel',data=df)

plt.show()

df.describe().transpose()

\_

plt.figure(figsize=(5,4))

vp = sns.violinplot(x='class',y='age',palette='rainbow',data=df)

plt.show()

\_

plt.figure(figsize=(5,4))

pq = sns.histplot(x='fare',bins=10,data=df,hue='survived',kde=False)

for i in pq.containers:

pq.bar\_label(i)

plt.show()

\_

plt.figure(figsize=(5,4))

st=sns.scatterplot(x='age',y='fare',data=df)

plt.show()

\_

plt.figure(figsize=(5,4))

kl=sns.scatterplot(x='age',y='fare',data=df,hue='survived')

plt.show()

**distplot in Seaborn**

The function sns.distplot() is part of the Seaborn library, which is used for visualizing the distribution of a dataset. It provides a way to see how data is distributed along a continuous scale, such as age, height, income, etc.

### What does sns.distplot() do?

* The function creates a **distribution plot** that shows the distribution of data through either a **histogram** or a **kernel density estimate (KDE)**, or both.
* By default, it plots both a histogram and a KDE line, so you can see how the data is spread and estimate its density.

**Why Use distplot?**

* It helps to visualize how your data is distributed (e.g., normally distributed, skewed, bimodal, etc.).
* It combines **histograms** (to see how many data points fall into specific ranges) and **KDEs** (to understand the general distribution in a smoother way).

Let's break down your provided code step-by-step to give you a detailed and proper explanation of what each part does.

**Step-by-Step Code Breakdown:**

import seaborn as sns

from matplotlib import pyplot as plt

* **seaborn (sns)**: A powerful Python library for creating statistical graphics. It's built on top of Matplotlib and provides a high-level interface for drawing attractive and informative visualizations.
* **pyplot (plt)**: A module from Matplotlib used for plotting graphs, such as histograms, scatter plots, etc.

df = sns.load\_dataset('titanic')

df.head()

* **sns.load\_dataset('titanic')**: This loads the **Titanic dataset** provided by Seaborn, which contains information about passengers aboard the Titanic. The dataset includes columns such as age, class, fare, survived, and others.
* **df.head()**: Displays the first 5 rows of the Titanic dataset so that you can quickly inspect the data and its structure.

plt.figure(figsize=(6,4))

sns.displot(df['age']) # Use sns.distplot(df['age']) for older versions of seaborn library

plt.show()

* **plt.figure(figsize=(6,4))**: Creates a new figure for the plot, and figsize=(6,4) sets the size of the figure to 6 inches by 4 inches.
* **sns.displot(df['age'])**: This function creates a **histogram** of the age column in the Titanic dataset. It shows the distribution of ages among the passengers.
  + **displot** is useful for showing univariate distributions. It automatically chooses the best type of plot (histogram, kernel density estimate, etc.) depending on the data type.
* **plt.show()**: Displays the plot on the screen.

plt.figure(figsize=(5,3))

bp = sns.boxplot(x='class', y='age', palette='pastel', data=df)

plt.show()

* **plt.figure(figsize=(5,3))**: Creates a figure with dimensions 5 inches by 3 inches.
* **sns.boxplot(x='class', y='age', palette='pastel', data=df)**: This creates a **boxplot**, which shows the distribution of age for each class of passengers (1st class, 2nd class, 3rd class).
  + **Boxplot**: It displays the **median**, **quartiles**, and potential **outliers** in the data. The box shows the interquartile range (IQR) with a line for the median, and whiskers show the range of the data.
  + The **palette='pastel'** argument specifies the color palette for the plot.
* **plt.show()**: Displays the plot on the screen.

df.describe().transpose()

* **df.describe()**: This function generates descriptive statistics for the numerical columns in the dataset (like age, fare, etc.). It includes values such as:
  + **Count**: Number of non-null values.
  + **Mean**: The average value.
  + **Std**: The standard deviation (measure of the spread of the data).
  + **Min** and **Max**: The smallest and largest values.
  + **25%, 50%, 75%**: The 25th, 50th, and 75th percentiles (quartiles).
* **.transpose()**: This transposes the result to make it more readable by showing columns as rows and vice versa.

plt.figure(figsize=(5,4))

vp = sns.violinplot(x='class', y='age', palette='rainbow', data=df)

plt.show()

* **plt.figure(figsize=(5,4))**: Creates a figure with dimensions 5 inches by 4 inches.
* **sns.violinplot(x='class', y='age', palette='rainbow', data=df)**: This creates a **violin plot**, which combines aspects of a boxplot and a density plot.
  + The **violin plot** shows the distribution of data (in this case, age) for each class (1st class, 2nd class, 3rd class). The width of the "violin" represents the density of the data points in that region, and the white dot in the middle of the "violin" represents the median.
  + **palette='rainbow'**: Specifies the color palette used for the plot.
* **plt.show()**: Displays the plot on the screen.

plt.figure(figsize=(5,4))

pq = sns.histplot(x='fare', bins=10, data=df, hue='survived', kde=False)

for i in pq.containers:

pq.bar\_label(i)

plt.show()

* **plt.figure(figsize=(5,4))**: Creates a figure with dimensions 5 inches by 4 inches.
* **sns.histplot(x='fare', bins=10, data=df, hue='survived', kde=False)**: This creates a **histogram** to show the distribution of fare prices, with the **hue='survived'** argument splitting the data into two groups (survived and not survived).
  + The **bins=10** argument specifies the number of bins for the histogram.
  + **kde=False**: Disables the Kernel Density Estimate (KDE) curve, which is a smoothed version of the histogram.
  + **hue='survived'**: The hue parameter colors the bars based on the survival status, so passengers who survived will have one color and those who did not will have another.
* **for i in pq.containers:**: This loop iterates through all the containers (bars) of the histogram.
  + **pq.bar\_label(i)**: Adds the value (count) at the top of each bar in the histogram.
* **plt.show()**: Displays the plot on the screen.

plt.figure(figsize=(5,4))

st = sns.scatterplot(x='age', y='fare', data=df)

plt.show()

* **plt.figure(figsize=(5,4))**: Creates a figure with dimensions 5 inches by 4 inches.
* **sns.scatterplot(x='age', y='fare', data=df)**: This creates a **scatter plot** to visualize the relationship between **age** and **fare**.
  + The x-axis shows age, and the y-axis shows fare. Each point on the plot represents a passenger, with the position determined by their age and fare.
* **plt.show()**: Displays the plot on the screen.

plt.figure(figsize=(5,4))

kl = sns.scatterplot(x='age', y='fare', data=df, hue='survived')

plt.show()

* **plt.figure(figsize=(5,4))**: Creates a figure with dimensions 5 inches by 4 inches.
* **sns.scatterplot(x='age', y='fare', data=df, hue='survived')**: This is a **scatter plot** similar to the previous one, but with the **hue='survived'** argument.
  + The **hue='survived'** parameter colors the points based on whether the passenger survived (survived=1) or not (survived=0). It helps you understand how survival status is related to age and fare.
* **plt.show()**: Displays the plot on the screen.

**Key Takeaways:**

1. **Histograms** (sns.histplot and sns.displot) show the distribution of a single variable.
2. **Boxplots** (sns.boxplot) visualize the spread and potential outliers in the data.
3. **Violin plots** (sns.violinplot) show data distribution with an added density estimate.
4. **Scatter plots** (sns.scatterplot) visualize relationships between two variables, and using **hue** allows coloring based on a third variable (like survival status).
5. **Bar Labels** are added to histogram bars to show the exact count of passengers in each bin.

This code is essentially helping you understand patterns in the Titanic dataset, such as how age and fare are distributed across passenger classes, and how survival status correlates with age and fare.

Here are some potential questions that could be asked during your viva (oral examination) for the code related to Titanic dataset visualizations:

**General Questions:**

1. **What is the Titanic dataset, and what information does it contain?**
   * You should be able to explain that it contains information about passengers on the Titanic, such as age, class, fare, survival status, and more.
2. **What do you understand by 'data visualization'? Why is it important?**
   * Explain that data visualization is the graphical representation of data to help understand trends, patterns, and insights quickly. It helps in making data-driven decisions and interpreting large datasets efficiently.
3. **Why did you choose Seaborn for this analysis?**
   * Discuss that Seaborn is a Python visualization library based on Matplotlib, which is easy to use, provides beautiful aesthetics, and integrates well with pandas data structures like DataFrames.
4. **Can you explain the purpose of sns.load\_dataset('titanic') in the code?**
   * This loads the Titanic dataset, which is a built-in dataset provided by Seaborn. It is used to test and demonstrate data visualization techniques.

**Questions Related to Specific Plots:**

1. **What is the purpose of the histogram (sns.displot) and how does it work?**
   * Explain that a histogram shows the distribution of numerical data. It groups data into bins (intervals) and plots the frequency of values in each bin. In this case, it's used to display the distribution of passenger ages.
2. **How would you interpret the results of the sns.displot(df['age']) plot?**
   * Describe the distribution of the age of Titanic passengers. Is it uniform, skewed, bimodal, etc.? Mention the peaks and the spread of data.
3. **What does the boxplot (sns.boxplot) show, and what insights can you gain from it?**
   * The boxplot shows the distribution of age for each class of passengers. It displays the median, interquartile range (IQR), and potential outliers. You can discuss the differences in age distribution across classes.
4. **What does the sns.violinplot do, and how is it different from a boxplot?**
   * A violin plot combines the boxplot and a kernel density estimate. It shows the data's distribution and density. Unlike the boxplot, it also indicates the distribution shape and spread, which gives a deeper understanding of data distribution.
5. **Explain the role of hue='survived' in the sns.histplot and sns.scatterplot.**
   * The hue parameter colors the data points based on a categorical variable, in this case, whether the passenger survived or not. This allows for comparison between groups visually.
6. **What is the significance of adding bar labels in the histogram?**

* The bar labels are used to display the exact counts of passengers in each bin, making it easier to interpret the distribution and compare the data directly.

**Technical and Code-related Questions:**

1. **What does plt.figure(figsize=(x, y)) do in the code?**

* This command creates a new figure for plotting and sets the size of the plot. The arguments (x, y) specify the width and height of the plot in inches.

1. **What does the sns.scatterplot(x='age', y='fare') do, and how can it be useful?**

* A scatter plot shows the relationship between two continuous variables (age and fare in this case). It can help identify correlations, trends, or outliers between the two variables.

1. **Can you explain the difference between sns.histplot and sns.boxplot?**

* sns.histplot is used for showing the distribution of a single variable by creating a histogram, while sns.boxplot shows the distribution summary (median, quartiles) along with potential outliers.

1. **Why do we use the plt.show() command after each plot?**

* plt.show() is used to display the plot in a new window. It ensures that the plot is rendered when using Matplotlib or Seaborn.

1. **What do you mean by 'kde=False' in sns.histplot?**

* This disables the Kernel Density Estimate (KDE) curve. The KDE would normally be used to smooth the histogram and show the probability density function, but in this case, it's turned off.

**Advanced Questions:**

1. **How could you improve or extend this analysis?**

* You could discuss adding more visualizations, comparing survival rates by class or gender, or performing statistical tests to analyze the differences in survival based on age or fare.

1. **What other types of plots can be used to visualize the age and fare distributions?**

* You could mention plots like **bar plots**, **pair plots**, **heatmaps**, or **correlation matrices**, which could provide different views of the data.

1. **What is the purpose of the sns.load\_dataset() function?**

* This function loads a built-in dataset from Seaborn. You can also use it to load other datasets, such as tips, iris, etc.

1. **What insights can you draw from the sns.scatterplot when hue='survived'?**

* Discuss how survival is related to age and fare, and how the hue feature visually distinguishes passengers who survived from those who didn’t.

1. **If you had missing values in the Titanic dataset, how would you handle them before visualization?**

* You can discuss various strategies like filling missing values with the mean, median, or mode, or dropping rows with missing data.

**Conceptual Understanding:**

1. **What is the difference between a boxplot and a violin plot?**

* A boxplot shows the summary of a dataset (e.g., median, IQR) and potential outliers, while a violin plot shows the same summary but also includes the data’s distribution shape (density).

1. **Why is it useful to visualize age and fare in relation to survival?**

* Visualizing these variables helps identify any patterns or correlations that can explain the survival likelihood, such as whether older passengers or passengers with higher fares had a better chance of survival.

1. **Can you explain how sns.violinplot uses colors?**

* In this case, the color palette helps distinguish between different classes, which gives insight into how age distributions differ across classes (e.g., 1st, 2nd, 3rd class).

By reviewing these questions and preparing answers, you'll be well-prepared for your viva. Good luck with your practical!

### 1. Types of Plots to Find Patterns in Data:

* **Histograms**: Show the distribution of a single continuous variable.
* **Boxplots**: Display the distribution of data and identify outliers.
* **Violin Plots**: Combine boxplots with density plots to show the distribution of data, especially for multiple categories.
* **Scatter Plots**: Show the relationship between two continuous variables.
* **Heatmaps**: Visualize correlation matrices or density of values.
* **Pair Plots**: Visualize relationships between multiple variables at once.
* **Bar Plots**: Used for comparing categorical data.
* **Line Plots**: Show trends over time or continuous data.
* **Swarm Plots**: Display individual data points for categorical data.

### 2. When to Use Distribution Plots vs Categorical Plots:

* **Distribution Plots** (e.g., histograms, KDE, displot):
  + **Used when**: You want to understand the distribution or frequency of a continuous variable.
  + **Examples**: Age distribution of passengers, income distribution, etc.
* **Categorical Plots** (e.g., boxplot, violin plot, bar plot):
  + **Used when**: You want to analyze data based on categorical variables (e.g., comparing the distribution of a continuous variable across categories).
  + **Examples**: Comparing the age distribution across different classes or survival rates.

### 4. Parameter to Add Another Categorical Variable to Violin Plot:

* **Parameter**: hue
  + **Purpose**: Adds another categorical variable to distinguish data within the same category (color-coding the data).
  + **Syntax**:
  + sns.violinplot(x='class', y='age', hue='survived', data=df, palette='rainbow')
  + **Explanation**:
    - x='class': Categories (e.g., 1st, 2nd, 3rd class).
    - y='age': The continuous variable to be plotted.
    - hue='survived': The additional categorical variable (e.g., survival status).
    - palette='rainbow': Color palette for different categories based on hue.