**import seaborn as sns**

**import matplotlib.pyplot as plt**

**from collections import Counter**

**\_**

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

**df.head()**

**\_**

**# Describe**

**print(df.describe())**

**# Describe - transposed, i.e. rows and columns swapped**

**print(df.describe().transpose())**

**\_**

**# Mean, median, mode**

**age\_data = df['age'].dropna() # Drop missing values in age & store in age\_data var**

**sorted\_age\_data = sorted(age\_data) # Store sorted age\_data**

**n = len(sorted\_age\_data) # Store length of age\_data**

**# Calculate mean**

**mean\_age = sum(age\_data) / len(age\_data)**

**# Calculate median**

**if n % 2 == 1: # odd**

**median\_age = sorted\_age\_data[n // 2]**

**else: # even**

**median\_age = (sorted\_age\_data[n // 2 - 1] + sorted\_age\_data[n // 2]) / 2**

**# Calculate mode**

**age\_counts = Counter(age\_data) # Count occurrences of each age**

**mode\_age = age\_counts.most\_common(1)[0][0] # Get the most common value**

**# Print**

**print(f"The mean age is: {mean\_age}")**

**print(f"The median age is: {median\_age}")**

**print(f"The mode age is: {mode\_age}")**

**\_**

**plt.figure(figsize=(8,4)) # 8 by 4 inches**

**sns.boxplot(x="sex", y="age", hue="survived", data= df, palette="viridis")**

**plt.title("Distribution of age with respect to each gender and survival Status")**

**plt.xlabel("Sex")**

**plt.ylabel("Age")**

**plt.show()**

**\_**

sns.violinplot(x='sex',y='age',data=df, hue= 'survived')

\_

sns.catplot(x="sex", hue="survived", data=df, kind="count")

\_

Let's go through the code step by step and explain what each part does.

**1. Importing Libraries**

import seaborn as sns

import matplotlib.pyplot as plt

from collections import Counter

* **seaborn (sns)**: This is a Python visualization library built on top of Matplotlib that makes it easier to create beautiful and informative statistical graphics. It's used here for plotting various types of graphs.
* **matplotlib.pyplot (plt)**: This is a library for creating static, animated, and interactive visualizations. It's used for controlling the figure size and displaying the plots.
* **Counter**: A part of the collections module in Python, it is used to count occurrences of elements. In this code, it is used to calculate the **mode** of the age data (the most frequently occurring age).

**2. Loading the Titanic Dataset**

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

df.head()

* **sns.load\_dataset('titanic')**: This loads the **Titanic dataset** directly from Seaborn's built-in datasets. It contains information about the passengers on the Titanic ship, including their age, gender, survival status, and class.
* **df.head()**: This shows the first 5 rows of the dataset to get a quick overview of the data.

**3. Descriptive Statistics**

# Describe

print(df.describe())

# Describe - transposed, i.e. rows and columns swapped

print(df.describe().transpose())

* **df.describe()**: This computes and displays summary statistics (like mean, std deviation, min, max, and quartiles) for each numeric column in the dataset.
* **df.describe().transpose()**: This is simply a transposed version of the above, where the rows and columns are swapped. This makes it easier to view statistics for each column as rows, especially when there are many columns.

**4. Calculating Mean, Median, Mode of Age**

# Mean, median, mode

age\_data = df['age'].dropna() # Drop missing values in age & store in age\_data var

sorted\_age\_data = sorted(age\_data) # Store sorted age\_data

n = len(sorted\_age\_data) # Store length of age\_data

# Calculate mean

mean\_age = sum(age\_data) / len(age\_data)

# Calculate median

if n % 2 == 1: # odd

median\_age = sorted\_age\_data[n // 2]

else: # even

median\_age = (sorted\_age\_data[n // 2 - 1] + sorted\_age\_data[n // 2]) / 2

# Calculate mode

age\_counts = Counter(age\_data) # Count occurrences of each age

mode\_age = age\_counts.most\_common(1)[0][0] # Get the most common value

# Print

print(f"The mean age is: {mean\_age}")

print(f"The median age is: {median\_age}")

print(f"The mode age is: {mode\_age}")

* **df['age'].dropna()**: This removes any missing or NaN values from the **age** column, as calculating mean, median, and mode requires complete data.
* **Calculating Mean**: The mean (or average) age is computed by summing all ages and dividing by the total number of passengers.
* **Calculating Median**: The median is the middle value when all the ages are sorted. If the number of data points is odd, it's the middle value; if even, it's the average of the two middle values.
* **Calculating Mode**: The mode is the most frequently occurring value in the **age** data. **Counter(age\_data)** counts the occurrences of each age, and most\_common(1)[0][0] gives the most frequent age.

**5. Box Plot**

plt.figure(figsize=(8, 4)) # 8 by 4 inches

sns.boxplot(x="sex", y="age", hue="survived", data=df, palette="viridis")

plt.title("Distribution of age with respect to each gender and survival Status")

plt.xlabel("Sex")

plt.ylabel("Age")

plt.show()

* **sns.boxplot()**: This creates a **box plot** to visualize the distribution of **age** for each **gender** (sex), and further splits by **survival status** (hue="survived").
  + **x="sex"**: The categories on the x-axis are **gender** (male or female).
  + **y="age"**: The numerical values on the y-axis represent **age**.
  + **hue="survived"**: This adds color coding based on survival status (whether the passenger survived or not).
  + **palette="viridis"**: This specifies the color palette for the plot.
* **The Box Plot**:
  + Displays the **median**, **interquartile range (IQR)**, and **outliers** of age for each gender, split by survival status.
* **plt.figure(figsize=(8, 4))**: This sets the size of the plot to 8 inches by 4 inches.

**6. Violin Plot**

sns.violinplot(x='sex', y='age', data=df, hue='survived')

* **sns.violinplot()**: This creates a **violin plot**, which combines aspects of both a box plot and a density plot. It shows the distribution of **age** for each gender, with additional color coding based on survival status (hue="survived").
  + The width of the plot shows how common certain ages are, and the **vertical line** in the center shows the **median** of the age distribution.

**7. Categorical Plot (Count Plot)**

sns.catplot(x="sex", hue="survived", data=df, kind="count")

* **sns.catplot()**: This is a high-level function used to create categorical plots.
  + **x="sex"**: We are plotting **gender** on the x-axis.
  + **hue="survived"**: This adds another layer of color, splitting the data based on whether the passenger survived or not.
  + **kind="count"**: This specifies a **count plot**, which will show the number of males and females who survived and those who did not survive.
* **Purpose**: This count plot will give a visual representation of the **number of survivors** and **non-survivors** for both **male** and **female** passengers.

**In summary:**

* **Data Loading**: We loaded the Titanic dataset.
* **Descriptive Statistics**: We computed summary statistics (mean, median, mode) of the dataset, particularly focusing on the age column.
* **Box Plot**: The **box plot** is used to visualize the distribution of ages for each gender and survival status.
* **Violin Plot**: The **violin plot** gives us an idea of the age distribution and how it varies between genders and survival status.
* **Categorical Count Plot**: The **count plot** visualizes the count of survivors vs non-survivors based on gender.

This code is useful to understand the relationships and distributions within the Titanic dataset using different types of plots.

**Expected questions for the viva:**Here are some possible viva questions related to your code and the concepts it involves:

### Questions Based on Code Implementation:

1. **What is the purpose of the sns.load\_dataset('titanic') method?**
   * This is used to load the Titanic dataset from Seaborn's inbuilt dataset library. Could you explain how it is structured?
2. **Explain the use of df.describe() and df.describe().transpose(). What is the difference between the two?**
   * df.describe() gives statistical information like count, mean, standard deviation, etc., for numerical columns.
   * df.describe().transpose() transposes this information (rows become columns, and columns become rows).
3. **What is the purpose of the dropna() method in df['age'].dropna()?**
   * dropna() is used to remove any missing or NaN values from the 'age' column before performing calculations.
4. **Can you explain how you calculated the mean, median, and mode for the 'age' column?**
   * Mean is calculated as the sum of the data divided by the count. Median is the middle value when the data is sorted. Mode is the most frequent value.
5. **What is the purpose of the Counter class from the collections module in this code?**
   * The Counter is used to count the frequency of occurrences of each value in the 'age' column and helps in calculating the mode.
6. **What is the purpose of the sns.boxplot() function, and how does it work in this context?**
   * sns.boxplot() is used to display the distribution of 'age' based on 'sex' and whether they survived ('hue="survived"'). It helps visualize the spread and skewness of the data.
7. **What does the hue='survived' parameter do in sns.boxplot() and sns.violinplot()?**
   * The hue='survived' parameter colors the plot based on the survival status, allowing us to distinguish between those who survived and those who didn’t.
8. **Explain the difference between a boxplot and a violinplot. When would you use one over the other?**
   * A boxplot shows the minimum, first quartile, median, third quartile, and maximum values. A violinplot combines a boxplot with a density plot to give a better understanding of the distribution. Use violinplot when you need to visualize the distribution in more detail.
9. **What does the sns.catplot() function do in the code?**
   * sns.catplot() is a categorical plot that helps visualize categorical data. Here, it shows the count of passengers who survived or did not survive, broken down by gender (sex).
10. **Explain the role of the palette="viridis" parameter in sns.boxplot()?**
    * The palette parameter defines the color scheme of the plot. In this case, the "viridis" palette is used, which is a perceptually uniform color palette.
11. **What would happen if you did not remove missing values using dropna() before calculating the mean, median, and mode?**
    * Missing values (NaNs) can affect the accuracy of calculations. For example, including them in the dataset might result in incorrect calculations or errors during computation.
12. **How would you interpret the results from the sns.violinplot() and sns.catplot() plots?**
    * The violin plot shows the distribution of age by sex, with additional detail on the survival status. The catplot displays the count of survivors for each gender.

### Questions Based on the Topic (Data Visualization):

1. **What are the common types of plots used in data visualization, and when would you use each type?**
   * Common types include histograms, box plots, scatter plots, line plots, bar plots, etc. You use a histogram for frequency distribution, a box plot for data spread, scatter plot for relationships, and so on.
2. **What is the purpose of using sns.boxplot() in data analysis?**
   * sns.boxplot() is used to show the distribution of numerical data based on categorical features, highlighting the central tendency, variability, and potential outliers.
3. **How do you handle missing data in your dataset?**
   * Missing data can be handled by removing missing values (dropna()), filling them with mean/median (imputation), or using algorithms that handle missing data.
4. **What is the significance of using hue in plots like boxplot and violinplot?**
   * The hue parameter allows us to split the data further based on categorical variables, adding another layer of information. In this case, it shows how survival status affects the distribution of age.
5. **What kind of insights can you derive from the combination of sns.boxplot() and sns.violinplot()?**
   * These plots help identify the spread, skewness, and outliers in the data. The boxplot gives you a high-level summary of the distribution, while the violin plot offers a more detailed distribution shape.
6. **What is the difference between a histogram and a countplot, and when would you use each?**
   * A histogram is used for continuous data and shows the frequency of data points in bins. A countplot is used for categorical data, showing the count of observations in each category.
7. **What does the sns.violinplot() reveal that a boxplot does not?**
   * A violin plot shows the probability density of the data at different values, giving more insight into the distribution’s shape, while the boxplot only shows the summary statistics.
8. **What do you understand by "Data Distribution" in the context of this Titanic dataset?**
   * Data distribution refers to how the values of the data (such as age, fare, etc.) are spread out or clustered. Understanding this helps identify patterns, outliers, and the central tendency of the data.

By preparing for these questions, you'll be better equipped to answer questions about both the code you've written and the concepts of data visualization you're using in the analysis.

Here are the answers to your questions:

**1. Explain Exploratory Data Analysis (EDA).**

**Exploratory Data Analysis (EDA)** is an approach to analyzing and summarizing datasets to gain a better understanding of the data’s structure, patterns, relationships, and any anomalies before performing more complex statistical modeling. EDA helps in the initial phase of data analysis and includes techniques like visualization, statistical graphics, and data transformations.

**Key steps in EDA:**

* **Data Cleaning:** Identifying and handling missing values, duplicates, and outliers.
* **Descriptive Statistics:** Calculating basic metrics like mean, median, mode, standard deviation, etc.
* **Data Visualization:** Using plots like histograms, scatter plots, box plots, and correlation heatmaps to visualize the relationships in the data.
* **Identifying Patterns:** Looking for trends, outliers, and correlations between variables.

**Purpose:** EDA helps to uncover hidden patterns, detect outliers, test assumptions, and verify the quality of the data before applying advanced techniques.

**2. Explain Univariate Analysis.**

**Univariate Analysis** refers to the analysis of a single variable or feature. The goal is to describe and summarize the data in terms of its distribution, central tendency (mean, median, mode), and dispersion (variance, standard deviation). It does not deal with relationships between multiple variables.

**Key Techniques for Univariate Analysis:**

* **Central Tendency Measures:** Mean, median, mode.
* **Dispersion Measures:** Range, variance, standard deviation.
* **Data Visualization:**
  + **Histograms:** Show the distribution of the data.
  + **Box Plots:** Visualize the spread and detect outliers.
  + **Bar Plots:** For categorical data, to count the number of occurrences of each category.

**Example:** When analyzing the "age" of passengers in the Titanic dataset, we would look at the mean, median, mode, and plot a histogram or box plot to visualize how the age of passengers is distributed.

**3. What is Multivariate Analysis? Explain.**

**Multivariate Analysis** refers to the analysis of more than one variable simultaneously to understand the relationships and interactions between them. Unlike univariate analysis, which focuses on a single variable, multivariate analysis looks at how multiple variables interact with each other.

**Key Techniques for Multivariate Analysis:**

* **Scatter Plots:** Show the relationship between two continuous variables.
* **Correlation Matrix:** Shows the correlation between multiple variables.
* **Multivariate Regression:** Modeling the relationship between multiple independent variables and a dependent variable.
* **Principal Component Analysis (PCA):** A dimensionality reduction technique to simplify large datasets while preserving their variance.
* **Pair Plots:** Display the pairwise relationships between several variables in a dataset.

**Example:** In the Titanic dataset, a multivariate analysis might examine how both age and fare correlate with survival, or how gender, age, and class together influence the likelihood of survival.

**4. How is Distplot and Boxplot created?**

**Distplot:** A **distplot** (deprecated in Seaborn version 0.11.0 and replaced by sns.histplot) is used to visualize the distribution of a continuous variable. It combines a histogram and a kernel density estimate (KDE) to show the frequency distribution of the data.

**Syntax for creating a distplot:**

import seaborn as sns

sns.displot(df['age'], kde=True) # kde=True adds the Kernel Density Estimate

**Explanation:**

* **df['age']** is the data to be plotted (in this case, the 'age' column from the Titanic dataset).
* **kde=True** adds a smooth curve that estimates the probability density function of the data.

The plot will display the histogram and a smooth KDE curve that helps in understanding the distribution of the data.

**Boxplot:** A **boxplot** is used to visualize the distribution of a continuous variable through its quartiles, highlighting the median, range, interquartile range (IQR), and potential outliers.

**Syntax for creating a boxplot:**

import seaborn as sns

sns.boxplot(x='sex', y='age', data=df, hue='survived', palette='viridis')

**Explanation:**

* **x='sex'**: The categorical variable that will be plotted on the x-axis (gender in this case).
* **y='age'**: The continuous variable plotted on the y-axis (age).
* **hue='survived'**: This is used to split the data based on whether passengers survived or not, adding another layer of information.
* **palette='viridis'**: Specifies the color palette to use for the plot.

The boxplot will show the minimum, first quartile (Q1), median, third quartile (Q3), and maximum values for age, and how they differ between genders and survival status. It also highlights any potential outliers outside the "whiskers" of the box.

**Summary of Key Points:**

* **Exploratory Data Analysis (EDA)** is about summarizing and visualizing datasets to understand patterns and anomalies before further analysis.
* **Univariate Analysis** focuses on analyzing one variable at a time.
* **Multivariate Analysis** explores the relationships between two or more variables.
* **Distplot** visualizes the distribution of a continuous variable and can include a KDE curve.
* **Boxplot** provides a summary of a variable's distribution, highlighting quartiles, median, and potential outliers.

These are fundamental concepts for performing data analysis and visualizing your findings, particularly when working with real-world datasets like Titanic.

**What is a catplot()?**

In Seaborn, **catplot()** is a **high-level function** used to create different types of categorical plots. Categorical plots are used when you're working with **categorical data** (like gender, color, or status). For example, when you want to compare the number of males and females or the survival rate of passengers.

catplot() is a versatile function because you can create different kinds of plots with it, such as:

* **Count Plot**
* **Box Plot**
* **Violin Plot**
* **Strip Plot**

**Basic Concept:**

* **Categorical Data**: Data that can be divided into distinct categories or groups (e.g., gender, age groups, regions).
* **Plotting**: Using catplot(), you can visualize these categories and compare them based on different factors, such as their counts or relationships with other variables.