

# Exp1 operation on dataset

06 May 2025 23:33

Data Wrangling | Perform the following operations using Python on any open source dataset (e.g., data.csv) 1. Import all the required Python Libraries. 2. Locate an open source data from the web (e.g., <https://www.kaggle.com>). Provide a clear description of the data and its source (i.e., URL of the web site). 3. Load the Dataset into pandas dataframe. 4. Data Preprocessing: check for missing values in the data using pandas isnull(), describe() function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame. 5. Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions. 6. Turn categorical variables into quantitative variables in Python. In addition to the codes and outputs, explain every operation that you do in the above steps and explain everything that you do to import/read/scrape the data set.

Here are **viva questions and answers** based on your full explanation of the Iris dataset preprocessing using pandas, NumPy, MinMaxScaler, LabelEncoder, and one-hot encoding:

## Viva Questions and Answers

### **Q1: Why do we use pandas in this code?**

A: We use pandas for reading the dataset, exploring the data, and performing preprocessing tasks like handling missing values and encoding.

### **Q2: What is the role of numpy here?**

A: numpy is used to identify numeric data types (np.number) and supports numerical operations used internally by other libraries.

### **Q3: What does MinMaxScaler do?**

A: MinMaxScaler scales all numeric values between 0 and 1. This is important for machine learning algorithms that are sensitive to data ranges.

### **Q4: Why do we use LabelEncoder?**

A: LabelEncoder is used to convert categorical labels like 'setosa', 'versicolor', and 'virginica' into numeric form (e.g., 0, 1, 2) for model input.

### **Q5: What is the purpose of get\_dummies()?**

A: pd.get\_dummies() performs one-hot encoding. It converts a categorical variable into separate binary columns, one for each category.

### **Q6: What is the difference between label encoding and one-hot encoding?**

A: Label encoding assigns a unique integer to each category. One-hot encoding creates separate columns for each category with 0/1 values.

### **Q7: Why do we create a copy of the DataFrame before scaling (df\_scaled = df.copy())?**

A: To preserve the original data and avoid overwriting it while scaling the numeric values.

### **Q8: What does df.describe(include='all') provide?**

A: It gives a statistical summary of all columns, including both numeric and categorical ones (like count, mean, std, top, freq, etc.).

### **Q9: What is the output of df.dtypes?**

A: It shows the data type of each column (e.g., int64, float64, object), helping us decide how to preprocess each column.

**Q10: What is the use of .shape in pandas?**

A: .shape returns a tuple (rows, columns), helping understand the size of the dataset.

**Q11: How do we select numeric and categorical columns separately?**

A: Using select\_dtypes(include=[np.number]) for numeric columns and select\_dtypes(include=['object', 'category']) for categorical columns.

**Q12: Why is feature scaling important in machine learning?**

A: Because models like KNN, SVM, and gradient descent-based algorithms perform poorly if features are on different scales.

**Q13: What does pd.concat([df\_scaled, df\_one\_hot], axis=1) do?**

A: It merges the scaled data and the one-hot encoded columns horizontally to form the final DataFrame.

**Q14: Why do we encode the Species column?**

A: Because most machine learning models can only work with numerical inputs, not strings or categories.

# Exp2 Academic\_performance

06 May 2025 23:41

Data Wrangling II Create an “Academic performance” dataset of students and perform the following operations using Python.

1. Scan all variables for missing values and inconsistencies. If there are missing values and/or inconsistencies, use any of the suitable techniques to deal with them.
2. Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them.
3. Apply data transformations on at least one of the variables. The purpose of this transformation should be one of the following reasons: to change the scale for better understanding of the variable, to convert a non-linear relation into a linear one, or to decrease the skewness and convert the distribution into a normal distribution. Reason and document your approach properly.

## Viva Questions and Answers

### 1. What is the purpose of the `fillna()` function in pandas?

**Answer:**

The `fillna()` function is used to fill missing values in a DataFrame. We can specify a value (such as mean, median, or mode) to replace the missing values in a particular column. In this case, we used `fillna()` to fill missing values in the Marks, Attendance, Age, and Gender columns. For example:

- `df['Marks'].fillna(df['Marks'].median(), inplace=True)` fills missing marks with the median value of that column.

### 2. Why do you use the median to fill missing values for the Marks column?

**Answer:**

The **median** is often used to fill missing values in numerical data, especially when the data has outliers or is skewed. The median is less affected by extreme values compared to the mean, making it a better representation of the central tendency when outliers are present. For example, in the Marks column, if there are extreme values like 200, using the median prevents those outliers from skewing the data.

### 3. What are outliers, and why is it important to handle them?

**Answer:**

Outliers are data points that differ significantly from the rest of the data. They can skew results, distort statistical analysis, and negatively impact models, especially those sensitive to extreme values (like linear regression). In the code, we detect outliers in the Marks column using the **Interquartile Range (IQR)** method. Any values above the upper limit of the IQR are considered outliers and are replaced with the median value of the column.

### 4. How does the IQR method help in detecting outliers?

**Answer:**

The **Interquartile Range (IQR)** is the range between the first quartile (Q1) and the third quartile (Q3). It helps in identifying the spread of the middle 50% of the data. Outliers are typically defined as values that fall below  $Q1 - 1.5 * IQR$  or above  $Q3 + 1.5 * IQR$ . These values are considered too extreme compared to the rest of the data and can be replaced or removed to avoid distortion in analysis.

In the code, we calculate the IQR for the Marks column and replace values greater than the upper limit with the median value.

### 5. What is the purpose of applying Min-Max Scaling on the CGPA

## **column?**

**Answer:**

Min-Max Scaling is a normalization technique used to rescale numeric data into a specified range, usually between 0 and 1. This scaling is essential when working with algorithms sensitive to the scale of the data, such as gradient descent-based algorithms or machine learning models. In the code, we apply Min-Max Scaling to the CGPA column to standardize its values, ensuring they are on a similar scale for further analysis or modeling.

## **6. Can you explain the difference between filling missing values with mean, median, and mode?**

**Answer:**

- **Mean:** The average of all values. It's suitable for normally distributed data but can be heavily influenced by outliers.
- **Median:** The middle value of a sorted dataset. It's less sensitive to outliers and is preferred when the data is skewed.
- **Mode:** The most frequent value. It's used for categorical data or when the most common value makes the most sense, as seen in the Gender column.

In our code:

- We used the **mean** for Attendance and Reading because these are numeric columns and expected to be normally distributed.
- We used the **median** for Marks because of potential outliers.
- We used the **mode** for Age and Gender to fill the most frequent values.

## **7. What is the significance of the inplace=True parameter in the fillna() method?**

**Answer:**

The `inplace=True` parameter modifies the original DataFrame in place, meaning the changes are directly applied to the DataFrame without creating a new object. Without it, the method would return a new DataFrame with the filled values, and the original DataFrame would remain unchanged. For example, `df['Marks'].fillna(df['Marks'].median(), inplace=True)` directly modifies the Marks column in df.

## **8. What is a Boxplot, and why is it used in this code?**

**Answer:**

A **Boxplot** is a graphical representation of the distribution of a dataset, showing the median, quartiles, and potential outliers. It helps visualize the spread of the data and the presence of any extreme values. In this code, we use a Boxplot to visualize the Marks column after handling outliers, making it easier to see the distribution of marks and confirm that the outliers have been addressed.

## **9. Why did you choose to use the mode to fill missing values in the Gender and Age columns?**

**Answer:**

The **mode** is suitable for categorical data, where the most frequent value is often the most logical choice. In the Gender column, filling missing values with the most frequent gender (mode) makes sense because gender is a categorical variable with only a few distinct values. Similarly, in the Age column, the mode represents the most common age, which can be a reasonable approximation for missing data in this case.

## **10. How would the results change if we had used the mean instead of the median for the Marks column?**

**Answer:**

Using the **mean** to fill missing values in the Marks column might distort the data if there are outliers.

For example, a value like 200 in the Marks column could skew the mean upward, leading to an overestimation of the typical mark. The **median**, on the other hand, is less affected by such extreme values and is therefore a better choice when outliers are present.

# Exp3 descriptiveStat\_statDetails\_eachType

06 May 2025 20:07

Descriptive Statistics - Measures of Central Tendency and variability Perform the following operations on any open source dataset (e.g., data.csv) 1. Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variable. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable. 2. Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of 'Iris-setosa', 'Iris-versicolor' and 'Iris-versicolor' of iris.csv dataset. Provide the codes with outputs and explain everything that you do in this step

Great! Here's a **complete set of viva questions and answers** based on the code you shared, covering both the tips and iris datasets, as well as related concepts. These are suitable for your **DSBDA viva exam preparation**.

## Viva Questions and Answers

### ◊ 1. What is the tips dataset?

**Answer:**

The tips dataset is a sample dataset provided by Seaborn. It contains information about restaurant bills, including total bill amount, tip, gender of customer, smoking status, day, time, and party size.

### ◊ 2. What is the purpose of copying the sex column into Gender?

**Answer:**

We copy the sex column into Gender to rename it for better clarity and then drop the original sex column. It helps in customizing or simplifying the column names.

### ◊ 3. Why do we use drop(columns="sex", inplace=True)?

**Answer:**

This removes the sex column from the DataFrame permanently (inplace=True), since we already copied its values to the new Gender column.

### ◊ 4. What does groupby('Gender')[['total\_bill']].agg(...).do?

**Answer:**

It groups the data based on gender and calculates statistical values like mean, median, min, max, and standard deviation for the total\_bill column for each group.

### ◊ 5. Why do we use .map({'Male': 0, 'Female': 1})?

**Answer:**

We convert categorical gender values to numeric format (Male = 0, Female = 1). This is often required for machine learning models that can't handle string values.

### ◊ 6. What is the iris dataset?

**Answer:**

The Iris dataset is a classic dataset in machine learning and statistics. It contains data about 3 types of iris flowers (setosa, versicolor, virginica) and their measurements like sepal and petal length/width.

## ◊ 7. How do you filter only 'setosa' species from the iris dataset?

Answer:

```
setosa = another_df[another_df['species'] == 'setosa']
```

This selects only the rows where species is 'setosa'.

## ◊ 8. What does .describe() do on a DataFrame?

Answer:

It returns descriptive statistics like count, mean, std, min, max, and quartiles for each numeric column in the DataFrame.

## ◊ 9. What does unique() do?

Answer:

The unique() function returns all unique values from a column. For example,

```
another_df['species'].unique()
```

[ 'setosa', 'versicolor', 'virginica' ]

## ◊ 10. What is the difference between mean() and median()?

Answer:

- **Mean** is the average of values.
- **Median** is the middle value when data is sorted.  
Median is better when the data contains outliers.

## ◊ 11. What type of variables are in the Iris dataset?

Answer:

- **Numerical columns**: sepal\_length, sepal\_width, petal\_length, petal\_width
- **Categorical column**: species

## ◊ 12. Can we visualize these datasets? How?

Answer:

Yes, we can use Seaborn or Matplotlib to visualize these datasets using plots like histograms, scatter plots, boxplots, etc.

## ◊ 13. Why is Seaborn used here?

Answer:

Seaborn is used to **load built-in datasets** (like tips and iris) and is also powerful for data visualization.

Part 1: Age Group, Income & Education Analysis using Pandas

### Q1: What is the purpose of using groupby() in this code?

A: groupby() is used to group the dataset by Age\_Group so we can calculate statistical summaries (mean, median, min, max, std) for each group's income.

### Q2: What does .agg(['mean', 'median', 'min', 'max', 'std']) do?

A: It applies multiple aggregation functions on the grouped data to get the mean, median, minimum, maximum, and standard deviation of Income per Age\_Group.

### Q3: Why is Age\_Group mapped to numeric values?

A: Because some machine learning models require numerical input. Converting categorical variables (20-30, 30-40, etc.) to numeric helps in further analysis or modeling.

### Q4: What is the use of map() function in this context?

**A:** The map() function replaces the Age\_Group string labels with corresponding numeric values as per the mapping dictionary.

**Q5: What data type is used to store initial data before creating the DataFrame?**

**A:** A Python dictionary is used, where column names are keys and values are lists.

## Part 2: Iris Dataset Analysis

**Q6: What is the Iris dataset?**

**A:** It is a classical dataset containing 150 samples of iris flowers, with 4 features (sepal length, sepal width, petal length, petal width) and 1 target column (Species) with 3 classes: Iris-setosa, Iris-versicolor, and Iris-virginica.

**Q7: What is the purpose of df1.describe()?**

**A:** It gives summary statistics (count, mean, std, min, max, quartiles) for all numerical columns.

**Q8: What does df1[df1['Species'] == 'Iris-setosa'].describe() do?**

**A:** It filters the dataset to only include rows where the species is Iris-setosa and then describes the numerical statistics of those rows.

**Q9: What does df1['Species'].unique() return?**

**A:** It returns a list of all unique species in the dataset: ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'].

**Q10: What is the purpose of df1.groupby(['Species']).mean()?**

**A:** It calculates the **average values of all numerical features** grouped by species, helping compare the average size of sepals/petals across species.

**Q11: What other statistical functions are used with groupby()?**

**A:** .mean(), .median(), and .std() — to get central tendency and spread of features by species.

**Q12: Why is it useful to describe each species separately?**

**A:** It helps in identifying patterns and differences among species, which is essential for classification problems in machine learning.

# Exp4 linearRegationusing\_Py\_R

06 May 2025 20:07

Data Analytics I Create a Linear Regression Model using Python/R to predict home prices using Boston Housing Dataset (<https://www.kaggle.com/c/boston-housing>). The Boston Housing dataset contains information about various houses in Boston through different parameters. There are 506 samples and 14 feature variables in this dataset. The objective is to predict the value of prices of the house using the given features.

## Simple Answer

### Question

What is Linear Regression?	A model that finds the best straight line to predict one variable (output) using another (input).
What are X and y here?	X is the input feature (number of rooms), and y is the target (house price).
Why do we split data into train and test?	To train the model on one part and test its accuracy on new, unseen data.
What is the intercept?	It's the starting value of y when X = 0.
What is the slope?	It tells how much the house price changes for each additional room.
What does the red line in the plot represent?	It's the predicted house prices using linear regression.
What if we use more than one feature in X?	Then it becomes Multiple Linear Regression.
Why use train_test_split?	To avoid overfitting and to test the model on new data.

## Basic Questions on Dataset and Problem

### Q1: What is the Boston Housing dataset?

**A:** It is a dataset that contains information about houses in Boston suburbs. It includes 506 samples and 14 attributes such as crime rate, average number of rooms, tax rate, and the target variable MEDV (Median value of owner-occupied homes in \$1000s).

### Q2: What is the objective of this practical?

**A:** The goal is to build a **Linear Regression model** to predict house prices (MEDV) using the given features.

### Q3: What is the shape of the dataset?

**A:** The dataset has **506 rows and 14 columns** (depending on whether any rows were dropped or not during preprocessing).

## Questions on Preprocessing

### Q4: How do you handle missing values in this dataset?

**A:**

- For numeric columns: Fill missing values with **mean**.
- For categorical column (CHAS): Use the **mode** (most frequent value).

### Q5: Why is it important to handle missing values?

**A:** Missing values can cause errors in model training and reduce model accuracy. Filling them ensures the dataset is complete.

## Questions on Model Building

### Q6: What is Linear Regression?

A: It's a supervised machine learning algorithm that models the relationship between a dependent variable and one or more independent variables using a straight line.

### Q7: What are the independent and dependent variables in your model?

A:

- **Independent variables:** All features except MEDV
- **Dependent variable:** MEDV (target)

### Q8: What is the formula for linear regression?

A:

$$y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

Where  $b_0$  is the intercept, and  $b_1, b_2, \dots, b_n$  are coefficients for the input features.

### Q9: How did you split the dataset for training and testing?

A: Using `train_test_split()`, 80% for training and 20% for testing.

## Questions on Model Evaluation

### Q10: How do you evaluate the performance of the regression model?

A: Using:

- **MSE (Mean Squared Error)**
- **RMSE (Root Mean Squared Error)**
- **R<sup>2</sup> Score (coefficient of determination)**

### Q11: What is R<sup>2</sup> Score?

A: It tells how well the model explains the variation in the target variable.

- $R^2 = 1$ : perfect prediction
- $R^2 = 0$ : model does no better than mean prediction

## Questions on Visualization

### Q12: What does the Actual vs Predicted plot show?

A: It shows how close the predicted house prices are to the actual prices.

- If points lie close to the red diagonal line, the predictions are accurate.

### Q13: What does the histogram of MEDV show?

A: It shows the distribution of house prices — how many houses fall in each price range.

## Conceptual & Practical Understanding

### Q14: Why do we use `train_test_split()`?

A: To evaluate the model on unseen data and prevent overfitting.

### Q15: Why is Linear Regression suitable for this problem?

A: Because the target variable (MEDV) is **continuous**, and we are trying to find a linear relationship between it and other numerical features.

### Q16: What will you do if the R<sup>2</sup> score is too low?

A: Try using:

- More features (instead of dropping columns)
- Polynomial regression or other models (e.g., Random Forest)
- Feature scaling or transformation

# Exp5 Implement logistic regression using Python/R

06 May 2025 23:54

Data Analytics II 1. Implement logistic regression using Python/R to perform classification on Social\_Network\_Ads.csv dataset. 2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

Here are the **viva questions and answers** in short and simple form:

## Viva Questions and Answers:

### 1. What is logistic regression?

- **Answer:** Logistic regression is a classification algorithm used to predict binary outcomes (0 or 1). It calculates probabilities using the sigmoid function.

### 2. Why do we use the sigmoid function in logistic regression?

- **Answer:** The sigmoid function maps predictions to values between 0 and 1, which represent probabilities for binary classification.

### 3. What is the purpose of the train\_test\_split function?

- **Answer:** It splits the data into training and testing sets, typically 75% for training and 25% for testing, to evaluate the model's performance.

### 4. Why is feature scaling important in logistic regression?

- **Answer:** Feature scaling ensures all features have the same scale, which helps the model learn efficiently and prevents some features from dominating others.

### 5. What does StandardScaler do?

- **Answer:** StandardScaler standardizes features by making them have a mean of 0 and a standard deviation of 1.

### 6. What is a confusion matrix?

- **Answer:** A confusion matrix shows how well the model performed by comparing the actual and predicted classes. It contains True Positives, False Positives, True Negatives, and False Negatives.

### 7. How do you calculate accuracy?

- **Answer:** Accuracy is the percentage of correct predictions:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

### 8. What is precision?

- **Answer:** Precision is the ratio of correct positive predictions to the total predicted positives:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

### 9. What is recall?

- **Answer:** Recall is the ratio of correct positive predictions to the total actual positives:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

### 10. What is error rate?

- **Answer:** Error rate is the percentage of incorrect predictions:

$$\text{Error Rate} = 1 - \text{Accuracy}$$

# Exp6 naïve bayes classification

06 May 2025 23:54

Data Analytics III 1. Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset. 2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

Here are **short and simple viva questions and answers** based on your Naive Bayes Iris dataset practical:

## ◊ Viva Questions and Answers

**Q1. What is Naive Bayes?**

A: Naive Bayes is a classification algorithm based on Bayes' Theorem. It assumes all features are independent.

**Q2. Why is it called “Naive”?**

A: Because it assumes that all features are independent, which is rarely true in real life.

**Q3. What is Gaussian Naive Bayes?**

A: Gaussian Naive Bayes assumes the features follow a normal (Gaussian) distribution.

**Q4. What type of problem is solved using the Iris dataset?**

A: It is a **multi-class classification** problem (three flower species).

**Q5. Why did you use LabelEncoder in this code?**

A: Because machine learning models can't handle text labels, so we convert species names to numbers.

**Q6. What does the confusion matrix tell us?**

A: It shows how many predictions were correct or incorrect for each class.

**Q7. What is accuracy?**

A: Accuracy is the ratio of correct predictions to total predictions.

**Q8. What is the difference between precision and recall?**

A:

- **Precision:** How many predicted positives are correct.
- **Recall:** How many actual positives are correctly predicted.

**Q9. What is the shape of the confusion matrix in multi-class classification?**

A: It is an  $N \times N$  matrix where  $N$  is the number of classes.

**Q10. Why do we use train\_test\_split?**

A: To separate data for training and testing, so we can evaluate model performance on unseen data.

# Exp7 Text Analytics

07 May 2025 00:09

7. Text Analytics 1. Extract Sample document and apply following document preprocessing methods: Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization. 2. Create representation of document by calculating Term Frequency and Inverse Document Frequency.

Here are the simplified and short versions of the viva questions and answers:

## Viva Questions and Answers

### **Q1: What is tokenization?**

**Answer:**

Tokenization is breaking text into smaller parts like sentences or words to make it easier to process.

### **Q2: What is the difference between stemming and lemmatization?**

**Answer:**

- **Stemming** removes prefixes or suffixes (e.g., "running" becomes "run").
- **Lemmatization** reduces words to their base form using a dictionary (e.g., "better" becomes "good").

### **Q3: Why do we remove stopwords?**

**Answer:**

Stopwords (like "the", "is") are removed because they don't add much meaning to the text and can be ignored for analysis.

### **Q4: What does re.sub() do in text processing?**

**Answer:**

re.sub() removes non-alphabetical characters (like punctuation) from text, keeping only the letters.

### **Q5: What is TF-IDF?**

**Answer:**

TF-IDF is a method to measure the importance of a word in a document, considering how often it appears and how rare it is in the whole dataset.

### **Q6: Why is TF-IDF useful in text analysis?**

**Answer:**

TF-IDF helps identify important words by reducing the weight of common words and highlighting unique ones.

### **Q7: Why use lemmatization instead of stemming?**

**Answer:**

Lemmatization gives valid words as results, while stemming may produce incorrect words (e.g., "countr" instead of "country").

### **Q8: What is nltk.download() used for?**

**Answer:**

nltk.download() downloads necessary resources like tokenizers and stopwords from NLTK for text processing.

### **Q9: What does the "Processed Corpus" output show?**

**Answer:**

It shows the cleaned and normalized text after removing stopwords and lemmatizing words, ready for further analysis.

### **Q10: How does TF-IDF help in text classification?**

**Answer:**

TF-IDF converts text into numbers that represent word importance, helping machine learning models classify text better.

# Exp8 Data Visualization I Titanic Dataset using Seaborn

07 May 2025 00:09

Here are some short and simple viva questions and answers based on the code and its output:

## **1. What is Seaborn, and why is it used here?**

**Answer:** Seaborn is a Python data visualization library built on top of Matplotlib. It provides an easier and more attractive way to create statistical plots, like histograms and boxplots. It's used here to visualize the Titanic dataset.

## **2. What is the purpose of the sns.histplot() function?**

**Answer:** The sns.histplot() function creates a histogram to show the distribution of data. In this case, it's used to plot the distribution of ticket fares and passenger ages.

## **3. What is a boxplot and how does it help in understanding the data?**

**Answer:** A boxplot is a graphical representation of the distribution of data based on a five-number summary: minimum, first quartile, median, third quartile, and maximum. It helps to identify the spread and outliers in the data, like the distribution of fares across passenger classes.

## **4. What is the purpose of using dropna() in the age distribution plot?**

**Answer:** The dropna() function removes any missing (NaN) values from the 'age' column to ensure that only valid age data is used for the histogram.

## **5. How does a KDE (Kernel Density Estimate) help in visualizing the data?**

**Answer:** A KDE is a smooth curve that represents the probability density function of the data. It helps in understanding the underlying distribution more clearly than a regular histogram.

## **6. What is the significance of the sns.barplot() function in this code?**

**Answer:** The sns.barplot() function creates a bar plot that shows the relationship between categorical variables (like passenger class) and a numerical variable (like survival rate). It is used here to show the survival rate across different passenger classes.

## **7. Why is the Titanic dataset useful for this type of analysis?**

**Answer:** The Titanic dataset is a popular dataset for learning data analysis and visualization techniques because it contains both categorical and numerical variables, allowing us to explore patterns and relationships, such as survival rates and class distribution.

## **8. What do you understand by the pclass column in the Titanic dataset?**

**Answer:** The pclass column represents the class of the passenger on the Titanic (1st class, 2nd class, or 3rd class). It helps in comparing survival rates and other features across different classes.

## **9. What does the survived column represent?**

**Answer:** The survived column indicates whether a passenger survived the Titanic disaster. It's a binary variable, where 1 means the passenger survived and 0 means the passenger did not survive.

## **10. What do you conclude from the survival rate by passenger class plot?**

**Answer:** From the survival rate by passenger class plot, we can observe that the survival rate was higher for 1st class passengers compared to 2nd and 3rd class passengers. This shows a disparity in survival chances based on class.

These questions cover the key concepts and output of your code, providing concise answers for viva preparation.

# Exp9 Data Visualization II – Titanic

## Dataset(boxPlot\_ageGender)

07 May 2025 00:14

Data Visualization II 1. Use the inbuilt dataset 'titanic' as used in the above problem. Plot a box plot for distribution of age with respect to each gender along with the information about whether they survived or not. (Column names : 'sex' and 'age') 2. Write observations on the inference from the above statistics.

### Simple Viva Questions and Answers

Question	Answer
What does sns.boxplot() do?	It shows the spread and distribution of data using boxes and whiskers.
What does the Titanic dataset contain?	Info about Titanic passengers like age, sex, class, fare, and survival.
What is hue used for in Seaborn plots?	It adds a color-based split to a plot based on a category, like survival.
What does the box in a boxplot represent?	Median, quartiles (IQR), and outliers of the data.
Why do we use plt.show()?	To display the final plot.
What is the use of legend in plots?	It tells what each color or style in the plot represents.
What insight can you gain from these plots?	Survival chances varied by gender and age. Female survivors had a different age distribution than males.

# Practical10: Data Visualization III – Iris Dataset

07 May 2025 00:15

10. Data Visualization III Download the Iris flower dataset or any other dataset into a DataFrame. (e.g., <https://archive.ics.uci.edu/ml/datasets/Iris>). Scan the dataset and give the inference as: 1. List down the features and their types (e.g., numeric, nominal) available in the dataset. 2. Create a histogram for each feature in the dataset to illustrate the feature distributions. 3. Create a boxplot for each feature in the dataset. 4. Compare distributions and identify outliers.

Here are the **viva questions & answers** (simple and short) and some possible **teacher modifications** for your Iris Data Visualization practical:

## Viva Questions & Answers (Simple & Short)

Question	Answer
Q1. What is the Iris dataset?	A dataset with 150 flowers and 3 species: Setosa, Versicolor, Virginica.
Q2. How many features are there?	4 numerical features + 1 categorical (species).
Q3. What does sns.load_dataset('iris') do?	Loads the inbuilt Iris dataset from seaborn.
Q4. What is a histogram used for?	To show the frequency distribution of values.
Q5. What is the use of kde=True in histplot()?	It adds a smooth curve (Kernel Density Estimation).
Q6. What is a boxplot?	A plot that shows spread, median, and outliers.
Q7. Why do we plot by species?	To compare how features differ between species.
Q8. What does plt.tight_layout() do?	Adjusts spacing between plots to avoid overlap.
Q9. What is the range of petal_length?	It varies by species but roughly 1.0 to 6.9.
Q10. Which features are best for species classification?	Petal length and petal width.

## Possible Teacher Modifications / Questions

Teachers often ask for small changes to test your understanding. Here are common ones:

Modification / Question	What to Do
Can you add scatter plots instead of histograms?	Use sns.scatterplot(x=..., y=..., hue='species', data=iris)
Add swarmplot or stripplot to show distribution?	Use sns.swarmplot() or sns.stripplot() by species.
Show correlation heatmap?	Use sns.heatmap(iris.corr(numeric_only=True), annot=True)
Label axes properly.	Add plt.xlabel() and plt.ylabel() to all plots.
Add titles to each subplot.	Use ax.set_title("Your title") in subplot code.
Change color palette.	Use sns.set_palette("Set2") or any other.
What is the difference between seaborn and matplotlib?	Seaborn is built on matplotlib and is better for stats plots.
Can you show outliers in petal length?	Point them out in the boxplot (dots outside whiskers).

Here are some potential **viva questions** with short answers for the provided code on data

visualization using the **Iris dataset**:

## 1. What is the purpose of using `sns.load_dataset('iris')`?

- **Answer:** It loads the built-in Iris dataset, which contains measurements of flowers' sepal and petal lengths and widths, along with the species of the flowers.

## 2. What does the `iris.info()` method do?

- **Answer:** It provides information about the dataset, including the number of rows, the data types of each column, and the number of non-null values.

## 3. What are histograms used for in data visualization?

- **Answer:** Histograms are used to visualize the distribution of numeric data. They show how the values of a variable are spread across different ranges (bins).

## 4. What is the significance of adding a KDE (Kernel Density Estimate) to the histogram?

- **Answer:** The KDE curve provides a smooth estimate of the distribution, which helps in identifying the underlying shape of the data (e.g., normal distribution) and makes the histogram visually smoother.

## 5. How are boxplots useful in this dataset?

- **Answer:** Boxplots help in identifying the spread of data and detecting outliers. They show the median, interquartile range (IQR), and outliers, which helps to understand the distribution of features by species.

## 6. What do the whiskers in a boxplot represent?

- **Answer:** The whiskers represent the range of the data, typically extending to 1.5 times the IQR from the quartiles. Data points outside this range are considered outliers.

## 7. Why is `plt.subplots(2, 2)` used in the code?

- **Answer:** It creates a 2x2 grid of subplots to arrange multiple plots (histograms or boxplots) in a single figure for better comparison.

## 8. What does `sns.boxplot()` do?

- **Answer:** The `sns.boxplot()` function creates a boxplot to visualize the distribution of data, showing the median, quartiles, and outliers. It can also be used to compare distributions across categories (e.g., species).

## 9. What is the purpose of `plt.tight_layout()`?

- **Answer:** It adjusts the spacing between subplots to prevent overlapping labels and ensure the plots fit within the figure area without cutting off any content.

## 10. How does the hue parameter work in Seaborn's `sns.boxplot()`?

- **Answer:** The hue parameter is used to split the data by a categorical variable (e.g., survived) within the plot, allowing comparison between different groups for each feature.

## 11. What do you learn from the boxplots of features by species?

- **Answer:** You can compare the distribution of each feature (e.g., petal length) across species, observe if any species have wider ranges, and identify potential outliers or differences in medians.

## **12. What is the benefit of using Seaborn over Matplotlib directly?**

- **Answer:** Seaborn provides a higher-level interface with simpler syntax, better styling, and built-in support for more complex statistical plots, which makes it easier to create attractive and informative visualizations.

## **13. What do the different colors in the boxplot represent?**

- **Answer:** In the boxplot, each color corresponds to a different category or group (e.g., different species). This allows easy visual differentiation of the distributions by group.