**Short Questions (1 Mark)**

1. Create a DataFrame from a dictionary with two columns, 'Name' and 'Age', where 'Name' contains ["Alice", "Bob"] and 'Age' contains [25, 30].
2. What function would you use to display the first five rows of a DataFrame, and why is it useful?
3. How can you check for missing values in a DataFrame?
4. Describe how to filter rows in a DataFrame based on a condition in one of its columns.
5. What method would you use to remove duplicate rows in a DataFrame?
6. What is the purpose of the merge function in Pandas?
7. How can you sort a DataFrame by multiple columns?
8. How do Pandas represent missing data in numeric and non-numeric data types?
9. Provide a code to check for missing values in each column of a DataFrame.
10. What does the fillna() function do, and why is it useful? Provide a code example.
11. Describe how the dropna() method works in Pandas.
12. Describe 'ffill' or 'bfill' method.
13. Write a code using Pandas to replace all NaN values in a DataFrame with zeros.
14. What is the purpose of the pd.concat() function in Pandas?
15. How do you concatenate two DataFrames df1 and df2 along rows? Provide code.
16. What parameter would you use in pd.concat() to ignore index values and create a new default integer index?
17. Describe how to concatenate DataFrames df1 and df2 along columns.
18. How can you perform an inner join using pd.concat()?
19. How would you merge two DataFrames, df1 and df2, based on a common column 'ID'?
20. What command is used to save a plot to a file in Matplotlib?
21. How can you save a plot as a PNG file named plot\_image.png?
22. Which argument can you use in plt.savefig to save a plot with a transparent background?
23. If you want to save a plot with a higher resolution, which parameter should you adjust in plt.savefig?
24. Write the command to plot the points (1, 2), (2, 4), (3, 6) as a simple line plot.
25. How can you display a line plot after plotting it with plt.plot()?
26. What does the command plt.plot(x, y, 'o-') do differently from plt.plot(x, y)?
27. How can you change the line color to red in a line plot?
28. Which argument in plt.plot() allows you to set the line style as dashed?
29. Write a command to create a blue dotted line plot using x and y data.
30. How can you make the line width thicker in a plot?
31. What command is used to set the x-axis range from 0 to 10 in a plot?
32. How can you limit the y-axis between -5 and 5 in a plot?
33. Which command is used to add a title to a plot?
34. Write a command to label the x-axis as "Time (s)".
35. How do you add a legend to a plot?

**Broad Questions**

**Question 1:**

**Combine Datasets with concat and append**

* Given two small DataFrames with columns for name, age, and score, use pd.concat to combine these datasets.
* Question: Which method, concat or append, would you choose for combining a large number of DataFrames and why?

**Concatenate NumPy Arrays**

* Create two 1-dimensional NumPy arrays: array1 with values [1, 2, 3] and array2 with values [4, 5, 6].
* Use np.concatenate to combine them into a single array.

**Concatenate with Different Join Types**

* Create two DataFrames: df1 with columns id, name, and score, and df2 with columns id, age, and city.
* Use pd.concat with both inner and outer join types to combine these DataFrames based on the id column.
* Question: What’s the difference between inner and outer joins when using pd.concat?

**Merge DataFrames**

* Given df1 and df2 from step 3, use pd.merge() to combine them on the id column.
* Experiment with different how parameters (left, right, inner, outer) and briefly explain the output of each.

**Groupby and Aggregate Data**

* Using a DataFrame representing sales data (columns: product, region, sales, and date), group the data by region and calculate:
  + The total sales for each region.
  + The average sales for each product.
* Question: Why might you use groupby when analyzing datasets with multiple categories?

**Query with DataFrame.query and eval**

* Using the sales data DataFrame, use query to find rows where sales are greater than a specified threshold (e.g., 1000).
* Use eval to calculate the total sales of all products in a specific region.

**Question 2:**

**Generate Data:**

* Use np.linspace to create an array x of 100 values between 0 and 10.
* Compute the sine (np.sin(x)) and cosine (np.cos(x)) of x.

**Create a Figure with Subplots**:

* Create a figure with two subplots: one for a simple line plot and another for a scatter plot.

**Line Plot Customizations**:

* In the first subplot, plot the sine function as a solid line and the cosine function as a dashed line.
* Customize line colors (color='blue' for sine and color='orange' for cosine).
* Set axes limits from 0 to 10 on the x-axis and -1.5 to 1.5 on the y-axis.
* Add labels to the x-axis and y-axis, a title for the plot, and a legend for the sine and cosine curves.

**Scatter Plot with Error Bars**:

* In the second subplot, plot a scatter plot of x vs. sin(x).
* Add error bars to each data point using plt.errorbar. Use a random normal distribution to generate small errors (e.g., yerr=0.4).
* Set a distinct color and shape (e.g., red squares) for the scatter plot.

**Save the Figure**:

* Save the entire figure as a .png file with the filename trigonometric\_plots.png.

**Question 3:**

**Data Loading and Handling Missing Values**

* Load the Iris dataset from Scikit-Learn.
* Simulate missing values: Introduce a few missing values randomly into the dataset. For example, set 5 random entries in the sepal length and petal length columns to NaN.
* Handle missing data: Use the mean of each column to fill in these missing values.

**Data Splitting and K-Fold Cross-Validation**

* Split the dataset into features (X) and target (y).
* Implement K-Fold Cross-Validation with 5 folds using a Decision Tree Classifier.
* Calculate evaluation metrics (accuracy, precision, recall, and F1 score) across all folds.

**Hyperparameter Tuning**

* Tune hyperparameters (max\_depth, min\_samples\_split) using GridSearchCV with Cross-Validation.
* Identify the best hyperparameters and save them for later use.

*[Same example but with KNN, SVC]*

**Question 4:**

You have the following data on monthly rainfall (in millimeters) in your city for the first four months of the year:

* January: 78 mm
* February: 92 mm
* March: 56 mm
* April: 89 mm

1. Using the data provided, create a Pandas Series and label each month accordingly.
2. Select the rainfall data for February using the Series as if it were a dictionary.

**Question 5:**

Suppose you have a list of dictionaries where each dictionary represents a student's data in a class, including their 'Name,' 'Age,' and 'Score' on a recent exam:

1. Construct a Pandas DataFrame using this list of dictionaries.
2. Demonstrate how to select only the 'Score' column from this DataFrame as if it were a NumPy array.

**Question 6:**

Imagine a store that sells different products. Each product has information on its 'Product Name,' 'Price' (in dollars), and 'Stock' (units available). Create a DataFrame with the following data:

|  |  |  |
| --- | --- | --- |
| Product Name | Price | Stock |
| Pen | 1.5 | 100 |
| Notebook | 3.0 | 50 |
| Eraser | 0.5 | 200 |

1. Write the code to create this DataFrame.
2. Show how you would access the 'Price' for 'Notebook' using DataFrame indexing, treating it like a two-dimensional array.

**Question 7:**

Consider a dataset that tracks employee information, including their names, departments, and years of experience. Some of the values for 'Years of Experience' are missing. The data is given as follows:

import pandas as pd

# Employee data with missing values

data = {

    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva'],

    'Department': ['HR', 'IT', 'Finance', None, 'Marketing'],

    'Years of Experience': [5, None, 8, 6, None]

}

employees\_df = pd.DataFrame(data)

1. Write code to detect the missing values in each column.
2. Drop rows where the 'Department' data is missing.
3. Fill in missing values in the 'Years of Experience' column with the average years of experience from the available data.

**Question 8:**

You are working with a dataset of product information where some entries are missing. The data includes 'Product Name,' 'Price,' and 'Discount' percentage. Some products do not have a 'Price' listed, and some are missing the 'Discount' percentage. The data is given below:

import pandas as pd

# Product data with missing values

data = {

    'Product Name': ['Pen', 'Notebook', 'Eraser', 'Marker', 'Highlighter'],

    'Price': [1.5, None, 0.5, 1.2, None],

    'Discount': [10, 5, None, 15, None]

}

products\_df = pd.DataFrame(data)

1. Detect and display the count of missing values in each column.
2. Use an appropriate method to fill in the missing 'Price' values with a constant value of 1.0.
3. Fill in the missing 'Discount' values with the median discount from the available data.

**Question 9:**

You have been given data about a small grocery store's sales figures across different categories. The data shows the number of units sold per item category, average unit price, and customer ratings. Use this data to create visualizations that represent various aspects of the sales and customer ratings.

# Sales data

categories = ['Fruits', 'Vegetables', 'Dairy', 'Bakery', 'Snacks']

units\_sold = [120, 200, 150, 90, 130]

avg\_price = [2.5, 1.5, 2.0, 1.8, 3.0]  # in dollars

customer\_ratings = [4.5, 4.2, 4.0, 3.8, 4.1]

Using the data above, complete the following tasks:

1. **Bar Plot**: Plot a bar chart showing units sold for each category. Add a title, x-axis label ("Categories"), and y-axis label ("Units Sold"). Also, include a legend if needed.
2. **Scatter Plot**: Create a scatter plot of units\_sold (on the x-axis) vs. avg\_price (on the y-axis). Add a title and labels for both axes.
3. **Histogram**: Plot a histogram of customer\_ratings to see the distribution of ratings. Add a title and label the x-axis as "Customer Ratings" and the y-axis as "Frequency".
4. **Pie Chart**: Create a pie chart to show the percentage of units sold by category. Add a title and a legend showing the category names.
5. **Box Plot**: Plot a box plot to visualize the spread and central tendency of customer\_ratings. Add a title.

Use appropriate colors, and ensure each plot has a title and labeled axes (where applicable).