

Pandas Practice Questions – 2

Pandas Series:

1. Create a Pandas Series from a list and calculate its absolute values using the `abs()` function.
2. Create a Pandas Series and use the `map()` function to replace all occurrences of one value with another.
3. Create a Series of numbers and add 5 to each element using the `add()` function.

Pandas DataFrame:

4. Create a DataFrame with 3 columns. Use the `add_prefix()` method to add the prefix "Col_" to each column name.
5. Create a DataFrame and calculate the covariance matrix using the `cov()` method.
6. Create a DataFrame and replace missing values in a column using the `bfill()` method.
7. Convert a specific column in a DataFrame to integers using the `astype()` method.
8. Use the `at` property to modify a specific value in a DataFrame.
9. Count the number of non-null values in each column of a DataFrame using the `count()` method.
10. Create a DataFrame and use the `all()` method to check if all values in a column are greater than a given number.
11. Create a DataFrame and use the `any()` method to check if any value in a column meets a condition.

Reading CSV:

12. Read a CSV file into a Pandas DataFrame. Display the first 10 rows using the head() method.
13. Load a CSV file into a DataFrame and check the data types of each column.
14. Load a CSV file, then drop any rows containing missing values and display the cleaned DataFrame.

Cleaning Data:

15. Create a DataFrame with missing values. Use Pandas to fill the missing values with the mean of the respective columns.
16. Create a DataFrame where some numeric values are stored as strings (e.g., "10"). Convert these to proper numeric format using the astype() function.
17. Create a DataFrame with incorrect values in a specific column. Identify and correct those values programmatically.
18. Create a DataFrame with duplicate rows. Remove all duplicates using the drop_duplicates() method.
19. Write a function to clean a DataFrame by filling missing data, fixing incorrect types, and removing duplicates.

Data Correlation:

20. Create a DataFrame with at least two numeric columns. Compute the correlation between these columns using corr().

Aggregation Functions:

21. Create a DataFrame and use the agg() method to calculate the sum and mean of numeric columns.
22. Create a DataFrame and use the agg() function to apply multiple aggregation operations (e.g., min, max, mean) on a specific column.
23. Create a DataFrame and use the agg() method with a custom aggregation function (e.g., calculate the range).

Data Manipulation:

24. Create a DataFrame and use the `add()` function to add two DataFrames element-wise.
25. Create a DataFrame and use `map()` to convert a column of categorical data to numerical labels.
26. Create a DataFrame and check the shape and dimensions using the `shape` and `ndim` attributes.

Miscellaneous:

27. Create a DataFrame and use the `abs()` function to return the absolute value of numeric columns.
28. Load a CSV file into a DataFrame, drop any rows that contain values that do not meet a specified format.
29. Create a DataFrame with missing values. Use `bfill()` to backfill the missing values.
30. Create a DataFrame and use `astype()` to change the data type of a column from float to int.