Pandas Practice Questions – 3

---

1. astype()

a. Convert the column age in a DataFrame to float64.

b. Change a Pandas Series of integers into strings using astype().

c. Convert a DataFrame containing numeric values into int32 datatype.

2. at property

a. Access the value in row 2 and column 'age' using the at property.

b. Update the value at a specific row and column using at property.

c. Use at to access and change multiple values in a DataFrame.

3. bfill()

a. Create a DataFrame with some missing values and fill them using bfill().

b. Demonstrate how bfill() works with multiple columns in a DataFrame.

c. Show how to limit the number of bfill() operations to one row.

4. columns property

a. Display the column names of a DataFrame using the columns property.

b. Rename columns in a DataFrame by directly modifying the columns property.

c. Add a prefix to all column names using columns.

5. combine()

a. Combine two DataFrames by comparing each element and selecting the larger element.

b. Create a function that performs a custom operation while combining two DataFrames.

c. Combine two DataFrames column-wise and handle NaN values using a custom function.

6. count()

a. Count the non-null entries in each column of a DataFrame.

b. Count non-null values row-wise in a DataFrame.

c. Use count() to count occurrences in a grouped DataFrame.

7. cov()

a. Compute the covariance between two columns in a DataFrame.

b. Calculate the covariance matrix for an entire DataFrame.

c. Use cov() to find the covariance between different DataFrames.

8. copy()

a. Create a deep copy of a DataFrame and modify it without affecting the original.

b. Make a shallow copy of a DataFrame and observe how changes affect both.

c. Modify a DataFrame after using copy(deep=False) and explain the behavior.

9. cummax()

a. Compute the cumulative maximum of a column in a DataFrame.

b. Apply cummax() row-wise across multiple columns.

c. Use cummax() to find the cumulative maximum of a Series containing both positive and negative values.

10. explode()

a. Create a DataFrame where one column contains lists, then use explode() to flatten it.

b. Apply explode() to a column with strings and lists mixed together.

c. Use explode() on a multi-index DataFrame.

---

11. ffill()

a. Create a DataFrame with missing values and forward fill the missing data using ffill().

b. Use ffill() on a specific column in a DataFrame.

c. Apply ffill() with a limit to how many values can be forward filled.

12. isna()

a. Check which values in a DataFrame are NaN using isna().

b. Replace all NaN values in a DataFrame with a specific value.

c. Combine isna() with sum() to count the total number of NaN values in a DataFrame.

13. items()

a. Use items() to iterate over columns in a DataFrame and print the name and data.

b. Modify values in a DataFrame using a loop with items().

c. Demonstrate how to use items() to perform operations on specific columns based on conditions.

14. nunique()

a. Find the number of unique values in each column of a DataFrame using nunique().

b. Use nunique() with a specific axis to count unique values in rows.

c. Find unique values in a subset of columns using nunique().

15. sample()

a. Use sample() to randomly select 3 rows from a DataFrame.

b. Select a random sample of 30% of rows from a DataFrame.

c. Use sample() with weights to select a biased sample from a DataFrame.

16. take()

a. Use take() to select specific rows by index from a DataFrame.

b. Demonstrate how to use take() with a negative index.

c. Use take() to select multiple rows from a Series based on their positions.

17. groupby()

a. Group a DataFrame by a column and calculate the sum for each group.

b. Use groupby() to group by multiple columns and calculate the mean.

c. Apply a custom aggregation function to each group using groupby().

18. round()

a. Round the values of a DataFrame to two decimal places.

b. Use round() on a Series to round values to the nearest integer.

c. Round specific columns in a DataFrame to different decimal places.