PandasAssignment

Q1. How do you load a CSV file into a Pandas DataFrame?

Ans:- To load a CSV (Comma Separated Values) file into a Pandas DataFrame, you can use the read\_csv() function provided by the Pandas library. Here is an example of how to use it:

import pandas as pd

df = pd.read\_csv('path/to/file.csv')

Q2. How do you check the data type of a column in a Pandas DataFrame?

Ans:- To check the data type of a column in a Pandas DataFrame, you can use the dtype attribute of the column.

Here is an example of how to check the data type of a column named "column\_name" in a Pandas DataFrame called df:

column\_type = df['column\_name'].dtype

print(column\_type)

This code snippet will print the data type of the specified column to the console.

Pandas data types include object (for string or mixed data types), int64 (for integer values), float64 (for floating-point values), bool (for boolean values), datetime64 (for datetime values), and timedelta (for timedeltas). Pandas also provides extensions to support categorical data, sparse data, and more.

Q3. How do you select rows from a Pandas DataFrame based on a condition?

Ans:- Suppose you have a DataFrame df with columns "name", "age", and "city". To select all rows where the "age" column is greater than 30, you can use the following code:

selected\_rows = df[df['age'] > 30]

You can select rows from a Pandas DataFrame based on a condition by using boolean indexing.

Here's an example:

Suppose you have a DataFrame df with columns "name", "age", and "city". To select all rows where the "age" column is greater than 30, you can use the following code:

selected\_rows = df[df['age'] > 30]

This creates a boolean mask using the condition df['age'] > 30, which evaluates to True for rows where the "age" column is greater than 30 and False otherwise. The boolean mask is then used to select the rows from the original DataFrame df.

You can also use other comparison operators such as == (equal), < (less than), <= (less than or equal to), >= (greater than or equal to), and != (not equal).

You can also combine conditions using logical operators such as & (and) and | (or). For example, to select all rows where the "age" column is greater than 30 and the "city" column is "New York", you can use the following code:

selected\_rows = df[(df['age'] > 30) & (df['city'] == 'New York')]

This creates a boolean mask that evaluates to True for rows where both conditions are true and selects the corresponding rows from the DataFrame.

Q4. How do you rename columns in a Pandas DataFrame?

Ans:- You can rename columns in a Pandas DataFrame using the rename() method. The rename() method allows you to change the names of one or more columns in place or to create a new DataFrame with the renamed columns.

Q5. How do you drop columns in a Pandas DataFrame?

Ans:- You can drop columns in a Pandas DataFrame using the drop() method. The drop() method allows you to remove one or more columns from a DataFrame.

Q6. How do you find the unique values in a column of a Pandas DataFrame?

Ans:- You can find the unique values in a column of a Pandas DataFrame using the unique() method. The unique() method returns an array of the unique values in the column.

Q7. How do you find the number of missing values in each column of a Pandas DataFrame?

Ans:- You can find the number of missing values in each column of a Pandas DataFrame using the isnull() method and the sum() method.

Q8. How do you fill missing values in a Pandas DataFrame with a specific value?

Ans:- You can fill missing values in a Pandas DataFrame with a specific value using the fillna() method. The fillna() method allows you to replace all NaN (missing) values in the DataFrame with a specified value.

Q9. How do you concatenate two Pandas DataFrames?

Ans:- You can concatenate two Pandas DataFrames using the concat() function. The concat() function allows you to concatenate two or more DataFrames vertically or horizontally.

Q10. How do you merge two Pandas DataFrames on a specific column?

Ans:- You can merge two Pandas DataFrames on a specific column using the merge() method. The merge() method allows you to combine two DataFrames into a single DataFrame based on a common column.

Q11. How do you group data in a Pandas DataFrame by a specific column and apply an aggregation function?

Ans:- You can group data in a Pandas DataFrame by a specific column and apply an aggregation function using the groupby() method.

The groupby() method allows you to group a DataFrame by one or more columns, and then apply a function to each group. You can use any aggregation function that is available in Pandas, such as sum(), mean(), max(), min(), count(), etc.

Q12. How do you pivot a Pandas DataFrame?

Ans:- You can pivot a Pandas DataFrame using the pivot() method. The pivot() method allows you to reshape a DataFrame by converting one or more columns into multiple columns.

Q13. How do you change the data type of a column in a Pandas DataFrame?

Ans:- You can change the data type of a column in a Pandas DataFrame using the astype() method. The astype() method allows you to cast a column to a different data type.

Q14. How do you sort a Pandas DataFrame by a specific column?

Ans:- You can sort a Pandas DataFrame by a specific column using the sort\_values() method. The sort\_values() method allows you to sort a DataFrame by one or more columns.

Q15. How do you create a copy of a Pandas DataFrame?

Ans:- You can create a copy of a Pandas DataFrame using the copy() method. The copy() method creates a deep copy of the DataFrame, including the data and the index.

Q16. How do you filter rows of a Pandas DataFrame by multiple conditions?

Ans:- You can filter rows of a Pandas DataFrame by multiple conditions using the & operator (for "and" conditions) and the | operator (for "or" conditions).

Q17. How do you calculate the mean of a column in a Pandas DataFrame?

Ans:- You can calculate the mean of a column in a Pandas DataFrame using the mean() method. The mean() method calculates the average value of the values in a column.

Q18. How do you calculate the standard deviation of a column in a Pandas DataFrame?

Ans:- You can calculate the standard deviation of a column in a Pandas DataFrame using the std() method. The std() method computes the standard deviation of the values in a column.

Q19. How do you calculate the correlation between two columns in a Pandas DataFrame?

Ans:- You can calculate the correlation between two columns in a Pandas DataFrame using the corr() method. The corr() method calculates the correlation between two columns using one of several correlation measures, such as the Pearson correlation coefficient.

Q20. How do you select specific columns in a DataFrame using their labels?

Ans:- You can select specific columns in a Pandas DataFrame using their labels by passing a list of column labels to the indexing operator ([]).

Q21. How do you select specific rows in a DataFrame using their indexes?

Ans:- You can select specific rows in a Pandas DataFrame using their indexes using the loc method. The loc method allows you to select rows by label (index) and column label.

Q22. How do you sort a DataFrame by a specific column?

Ans:- You can sort a Pandas DataFrame by a specific column using the sort\_values() method. The sort\_values() method allows you to sort a DataFrame by one or more columns.

Q23. How do you create a new column in a DataFrame based on the values of another column?

Ans:- You can create a new column in a Pandas DataFrame based on the values of another column using the indexing operator ([]) and the desired operation.

Q24. How do you remove duplicates from a DataFrame?

Ans:- You can remove duplicates from a Pandas DataFrame using the drop\_duplicates() method. The drop\_duplicates() method allows you to remove duplicate rows based on one or more columns.

Q25. What is the difference between .loc and .iloc in Pandas?

Ans:-

The main difference between .loc and .iloc in Pandas is in how they are used to index and select data from a DataFrame:

* .loc is used to select rows and columns by their labels. The .loc indexer takes two arguments separated by a comma. The first argument is the row label or a list of row labels, and the second argument is the column label or a list of column labels.
* .iloc is used to select rows and columns by their integer positions. The .iloc indexer takes two arguments separated by a comma. The first argument is the row position or a list of row positions, and the second argument is the column position or a list of column positions.