PANDAS ASSIGNMENT

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

Ans. 1. Load the CSV into a DataFrame: import pandas as pd. df = pd.read_csv('data.csv')

2.Print the DataFrame without the to_string() method: import pandas as pd. ...

3.Check the number of maximum returned rows: import pandas as pd. ...

4,Increase the maximum number of rows to display the entire DataFrame: import pandas as pd.

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

Ans. To check the data type in pandas DataFrame we can use the “dtype” attribute. The attribute returns a series with the data type of each column. And the column names of the DataFrame are represented as the index of the resultant series object and the corresponding data types are returned as values of the series object.

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

Ans. You can use the `.loc` indexer to select rows based on a condition. For example: `df.loc[df['column_name'] == some_value]` This will return all rows where the value of the 'column_name' column is equal to some_value.

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

Ans. You can rename columns of a Pandas DataFrame using the DataFrame.rename() method. The syntax for this method is: DataFrame.rename(columns={oldname: newname}) For example, if you wanted to rename the column 'oldname' to 'newname', you would use the following code: df.rename(columns={'oldname': 'newname'}, inplace=True)

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

Ans. You can drop columns in a Pandas DataFrame by using the DataFrame.drop() method, passing in the column name or list of column names as the first argument, and setting the axis argument to 1.

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

Ans. You can use the pandas.DataFrame.unique() method to find the unique values in a column of a Pandas DataFrame. This method returns a NumPy array of the unique values in the specified column.

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

Ans. You can use the Pandas DataFrame's isnull() method in combination with the sum() method to find the number of missing values in each column.

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

Ans. You can use the fillna() method to fill missing values in a Pandas DataFrame with a specific value. The syntax for this is: dataframe.fillna(value) Where value is the value you want to use to fill the missing values.

Q9. How do you concatenate two Pandas DataFrames?

Ans. You can concatenate two Pandas DataFrames by using the pd.concat() function. This function will take in two DataFrames as arguments and return a single DataFrame that combines the two. For example, if you have two DataFrames df1 and df2, you can concatenate them together with the following code: pd.concat([df1, df2])

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

Ans.To merge two Pandas DataFrames on a specific column, you can use the pandas.DataFrame.merge() method. This method allows you to specify the column on which the two DataFrames should be joined, as well as the type of join (e.g. left, right, inner, outer).

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 by using the groupby() method. For example, to group data by 'column_name' and apply the sum() aggregation function, you would use the following syntax: df.groupby('column_name').sum()

Q12. How do you pivot a Pandas DataFrame?

Ans. You can pivot a Pandas DataFrame using the DataFrame.pivot() function. This function will take the column values and turn them into column headings. The syntax is as follows: DataFrame.pivot(index, columns, values) Where "index" is the column you want to use as the row labels, "columns" is the column you want to use as the column labels, and "values" is the column you want to populate the DataFrame with.

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 by using the astype() function. For example, if you want to change the data type of a column named 'column_name' to string, you can use the following code: df['column_name'] = df['column_name'].astype(str)

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

Ans. To sort a Pandas DataFrame by a specific column, you can use the DataFrame.sort_values() method. This method takes in the column name as the argument. The default sorting order is ascending, but you can also specify the sorting order as descending.

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

Ans. You can create a copy of a Pandas DataFrame by using the copy() method. For example: df2 = df.copy() This will create a new DataFrame, df2, that is a copy of the original DataFrame, df.

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

Ans. You can use the Pandas DataFrame.query() method to filter rows of a Pandas DataFrame by multiple conditions. For example: df.query('condition1 & condition2 & condition3') This will return only the rows where the three conditions all evaluate to True.

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 by using the mean() function. For example, if you have a DataFrame called df, you can calculate the mean of the column 'A' by using the following code: df['A'].mean()

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 DataFrame.std() method. The syntax is as follows: DataFrame.std(axis=0, skipna=None, level=None, ddof=1, numeric_only=None, **kwargs) Where axis = 0 (columns) and axis = 1 (rows).

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

Ans. The Pandas DataFrame method corr() can be used to calculate the correlation between two columns in a DataFrame. To calculate the correlation, call the corr() method on the DataFrame and pass in the column names of the two columns you wish to calculate the correlation between. For example: df.corr(['column_1', 'column_2'])

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

Ans. You can use the DataFrame's "loc" method to select specific columns in a DataFrame using their labels. For example, if you wanted to select columns labeled 'A' and 'B' from a DataFrame called df, you would use the following code: df.loc[:, ['A', 'B']]

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

Ans .You can use the `.loc` indexer to select specific rows in a DataFrame using their indexes. For example, if you wanted to select the rows with indexes 0, 1, and 2 from a DataFrame named `df`, you could use the following code: `df.loc[[0,1,2]]`

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

Ans. You can sort a DataFrame by a specific column using the sort_values() method. For example, to sort the DataFrame by the "Name" column in descending order, you can use the following code: df.sort_values(by='Name', ascending=False)

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 DataFrame based on the values of another column by using the .assign() method. For example, if you wanted to create a new column in a DataFrame called 'ratio' based on the values of a column called 'numerator', the following code could be used: df = df.assign(ratio=lambda x: x.numerator / x.denominator)

Q24. How do you remove duplicates from a DataFrame?

Ans. There are several ways to remove duplicates from a DataFrame. One way is to use the DataFrame.drop_duplicates() method. This method will return a new DataFrame with duplicate rows removed. Another way to remove duplicates is to use the

DataFrame.drop_duplicates(subset=None, keep='first', inplace=False) method. This method allows you to specify a subset of columns to check for duplicates. If a row has the same value in all of the specified columns, the row will be considered a duplicate and will be dropped. The keep parameter allows you to specify whether to keep the first or last encountered duplicate row, or to keep all duplicates. Finally, if you want to keep certain duplicates, you can use the DataFrame.duplicated() method to find and mark the duplicates, and then use the DataFrame.loc() method to access and filter the marked duplicates.

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

Ans. The primary difference between .loc and .iloc in Pandas is that .loc is label-based, which means that you can use labels such as column names to access data, while .iloc is integer-based, which means that you use numerical indices to access data. .loc is primarily used for selecting rows and columns by label, while .iloc is used for selecting rows and columns by position.