Documentation Agent - Chat with Any Repo

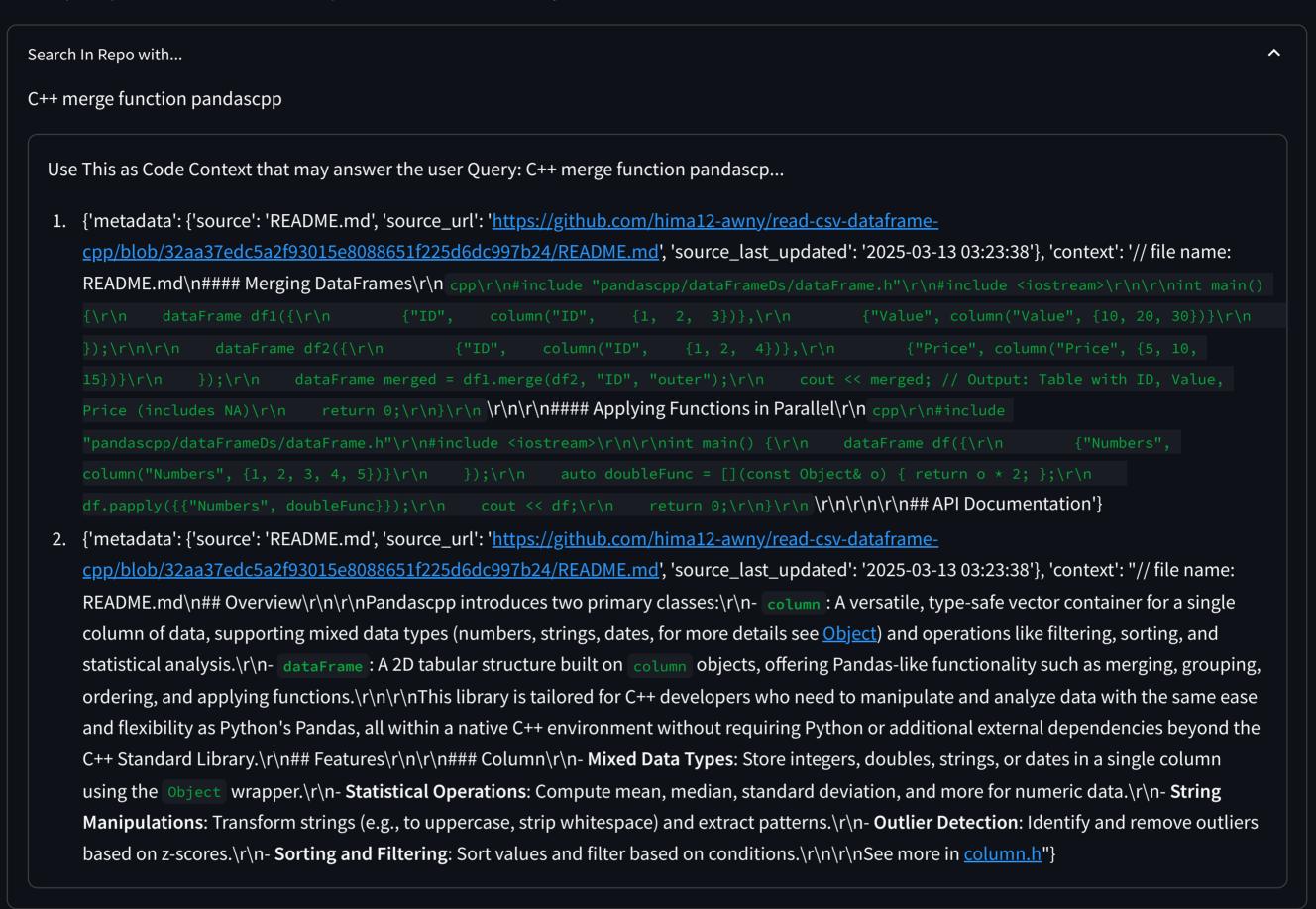


Chat with our AI assistant to get information about your Github Repo.

how merge function do his job?



I need more context to understand which merge function you are referring to. Could you please specify the file or library where this function is located? This will help me provide a more accurate explanation of its functionality.



The merge function in the dataFrame class of the pandascpp library combines two data frames based on a common column. Here's how it works, according to the README.md file:

- 1. Input: The merge function takes two data frames (dfl and df2), a common column name ("ID" in the example), and a merge type ("outer" in the example) as input.
- 2. **Functionality:** It merges the two data frames based on the specified column. The "outer" merge type includes all rows from both data frames, filling in missing values with NA where there is no match.
- 3. **Output:** The result is a new data frame (merged) that combines columns from both input data frames, aligned by the common column.

For example:

```
dataFrame df1({
             column("ID", {1, 2, 3})},
    {"Value", column("Value", {10, 20, 30})}
dataFrame df2({
             column("ID", {1, 2, 4})},
   {"Price", column("Price", {5, 10, 15})}
dataFrame merged = df1.merge(df2, "ID", "outer");
cout << merged; // Output: Table with ID, Value, Price (includes NA)</pre>
```

In this example, dfl and df2 are merged based on the "ID" column using an outer merge. The resulting merged data frame will contain all IDs from both dfl and df2, with corresponding "Value" and "Price" columns. If an ID exists in one data frame but not the other, the missing value will be filled with NA.

Sources



give me the flow of the implementation of it

I need to inspect the source code to provide you with the implementation flow of the merge function. I will generate a search query to retrieve the relevant code from the dataFrame.cpp file.

- 0;\r\n\r\n\tfor (const string& columns) {\r\n\t\tnew_df_data[colname] = column();\r\n\t\tnew_df_data[colname].mixed_type = data[colname].mixed_type;\r\n\t\tnew_df_data[colname].set_name(data[colname].name);\r\n\t\tall_cols[i++] = colname;\r\n\t}\r\n\tfor (const string& colname : other.columns) {\r\n\t\tif (colname == on) continue;'}

 2. {'metadata': {'source': 'dataFrame.h', 'source_url': 'https://github.com/hima12-awny/read-csv-dataframe
 - cpp/blob/32aa37edc5a2f93015e8088651f225d6dc997b24/pandascpp/dataFrameDs/dataFrame.h,' 'source_last_updated': '2025-03-13 03:23:38'}, 'context': '// file name: dataFrame.h\n> cols_conditions\r\n\t);\r\n\t#pragma endregion\r\n\r\n\t#pragma region Data Exploratory\r\n\r\n\t// group dataframe by columns names, and apply basic statics function in target column.\r\n\tdataFrame groupBy(\r\n\t\tvector<string> by,\r\n\t\tstring\t targetcol,\r\n\t\tStatFun func\r\n\t\);\r\n\r\n\t// order the dataframe by columns, sorting ascending by defaults.\r\n\t// enable revers, if revers with size 1 so apply this revers option for all by columns.\r\n\t// is not, must be the same length of the by columns.\r\n\tdataFrame orderBy(\r\n\t\tvector<string> by,\r\n\t\tvector<bool> revers = { false }\r\n\t);\r\n\r\n\t// merge two dataframes based on column name, \r\n\t// and the how to merge them Options = ['inner', 'outer', 'left', 'right']\r\n\tdataFrame merge(const dataFrame& other, const string on, string how = "inner");\r\n\r\n\t\#pragma endregion\r\n\r\n\t\#pragma region Apply\r\n\r\n\t// apply function to columns as key and their functions as value in input map.\r\n\tdataFrame apply(map<const std::string, std::function<Object(const Object&)>> col_functions);\r\n\t\#pragma endregion\r\n\r\n\t// this optimize the memory/size of the dataframe for each column. \r\n\tvoid optimiz_mem(bool with_mixed_types = true);\r\n\r\n\t/\n\t// convert/save the dataframe into csv formate with path to save to it.\r\n\tvoid to_csv(const std::string& targetPath);'}

The merge function in dataFrame.cpp combines two data frames (this and other) based on a common column (on) and a specified merge type (how). Here's a step-by-step breakdown of its implementation:

1. Get Merge Values:

o It calls <code>get_merg_df_values_on</code> for both data frames (<code>this and other</code>) to create unordered maps (<code>left_values and right_values </code>). These maps store the values of all columns <code>except</code> the <code>on column</code>, indexed by the values in the <code>on column</code>. This allows for efficient lookup during the merge process.

2. Initialize New Data Frame:

- It creates a new unordered_map called new_df_data to store the data for the merged data frame. This map will hold column objects for each column in the merged data frame.
- o It creates a vector<string> called all_cols to store the names of all columns in the merged data frame. The size of this vector is the sum of the number of columns in both input data frames (minus 1, to account for the shared on column).
- o It iterates through the columns of the this data frame and copies the column data and metadata (like mixed_type and name) to new_df_data. It also populates the all_cols vector with these column names.
- o It iterates through the columns of the other data frame. If a column name is not the on column and not already present in new_df_data, it creates a
 new column in new_df_data and adds the column name to all_cols.

3. Iterate Through 'On' Values and Merge:

- o The code iterates through the left_values map (representing the on column values from the this data frame). For each on value, it checks if the same on value exists in the right_values map (representing the on column values from the other data frame).
- Based on the how merge type (e.g., "inner", "outer", "left", "right"), it decides whether to include the current row in the merged data frame.
- o If the on value exists in both left_values and right_values (or if the merge type allows including the row even if it's only in one data frame), it adds the corresponding data from both data frames to the new_df_data columns.

4. Create and Return New Data Frame:

• After processing all on values, it constructs a new dataFrame object using the new_df_data map. The columns member of the new data frame is set to all_cols.

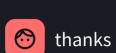
>

• Finally, it returns the newly created merged dataFrame.

In summary, the merge function efficiently combines data from two data frames based on a common column, handling different merge types to include or exclude rows based on the presence of matching values in the on column.

Sources





You're welcome! If you have any more questions about the merge function or anything else related to the pandascpp library, feel free to ask.

Type your message here.