Data Frame

0.1

Generated by Doxygen 1.9.1

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 DF Namespace Reference

Classes

- class DataFrame

  DataFrame is class for parsing data in a given file with a give delimeter (default is comma ','). all data will be saved as string wich user can later covert to desired type.

4.2 std Namespace Reference

Namespaces

- shorts

  namespace for introducting shortnames

4.3 std::shorts Namespace Reference

namespace for introducting shortnames

Typedefs

- using V_string = vector< string >
- using VV_string = vector< V_string >
- using Data = unordered_map< string, V_string >
- using V_double = vector< double >
- using V_int = vector< int >

4.3.1 Detailed Description

namespace for introducting shortnames

4.3.2 Typedef Documentation

4.3.2.1 Data

```
using std::shorts::Data = typedef unordered_map<string, V_string>
```

Definition at line 33 of file ReadFiles.hpp.

4.3.2.2 V_double

```
using std::shorts::V_double = typedef vector<double>
```

Definition at line 34 of file ReadFiles.hpp.

4.3.2.3 V_int

```
using std::shorts::V_int = typedef vector<int>
```

Definition at line 35 of file ReadFiles.hpp.

4.3.2.4 V_string

```
using std::shorts::V_string = typedef vector<string>
```

Definition at line 31 of file ReadFiles.hpp.

4.3.2.5 VV_string

```
using std::shorts::VV_string = typedef vector<V_string>
```

Definition at line 32 of file ReadFiles.hpp.

Chapter 5

Class Documentation

5.1 DF::DataFrame Class Reference

DataFrame is class for parsing data in a given file with a give delimeter (default is comma ','). all data will be saved as string wich user can later covert to desired type.

`#include <ReadFiles.hpp>`

Public Member Functions

- int get_n_rows () const

  Get the number of rows of a given data.
- int get_n_cols () const

  Get the number cols of a given data.
- void set_headers (std::shorts::V_string const &v_hdrs)

  Set the headers with user provided vector of strings.
- std::shorts::V_string get_headers () const

  Get the headers.
- std::shorts::V_string get_by_headers () const
- DataFrame copy () const

  to copy current data into new data frame
- DataFrame copy_by_hdrs (std::shorts::V_string const &v_hdrs)

  copy the requested data by provided headers name as vector of strings into a new data frame
- void remove_duplications (std::string const &hdr)

  get a given header and only keep the first occurance and remove the remaning rows
- void read_files (std::string_view path, char delim=',', bool is_first_col_header=true, std::shorts::V_string v_hdrs={})

  read files
- void read_files (std::string_view path, std::shorts::V_int const &v_cols_length, bool is_first_col_header=true, std::shorts::V_string v_hdrs={})
- void head (unsigned long long n=5)

  print n first rows off all columns

Public Attributes

- std::shorts::Data data

  saving data in an unrodered map
- unsigned long long n_rows

  number of rows
- unsigned long long n_cols

  number of columns
- std::shorts::V_string headers

  Header of data, whether is the first line or will be provided by user in case of user not providing header it would be string number start from one to number of columns of the data.
- std::unordered_map< int, int > mising_values

  an unorderd maps for missing values

Private Member Functions

- std::shorts::V_string read_lines (std::string_view path)
- void fill_data (std::shorts::V_string const &v_strs, char delim=',', bool is_first_col_header=true, std::shorts::V_string v_hdrs={})
- void fill_data (std::shorts::V_string const &v_strs, std::shorts::V_int const &v_cols_length, bool is_first_col↩ _header=true, std::shorts::V_string v_hdrs={})

5.1.1 Detailed Description

DataFrame is class for parsing data in a given file with a give delimeter (default is comma ','). all data will be saved as string wich user can later covert to desired type.

Definition at line 47 of file ReadFiles.hpp.

5.1.2 Member Function Documentation

5.1.2.1 copy()

```
DataFrame DF::DataFrame::copy ( ) const
```

to copy current data into new data frame

Returns

DataFrame new data frame

5.1.2.2 copy_by_hdrs()

```
DataFrame DF::DataFrame::copy_by_hdrs (
            std::shorts::V_string const & v_hdrs )
```

copy the requested data by provided headers name as vector of strings into a new data frame

Parameters

v_hdrs	

Returns

DataFrame new data frame

5.1.2.3 fill_data() [1/2]

```
void DF::DataFrame::fill_data (
            std::shorts::V_string const & v_strs,
            char delim = ',',
            bool is_first_col_header = true,
            std::shorts::V_string v_hdrs = {} )  [private]
```

Definition at line 46 of file ReadFiles.cpp.

```
47 {
48     std::shorts::VV_string vv_strs;
49 
50     for(auto const& line : lines)
51     {
52         std::istringstream iss(line);
53         std::string cell;
54         std::shorts::V_string v_str_tmp;
55 
56         while(std::getline(iss, cell, delim))
57         {
58             cell.erase(std::remove(cell.begin(), cell.end(), '\"'), cell.end());
59             v_str_tmp.emplace_back(cell);
60         }
61         vv_strs.emplace_back(v_str_tmp);
62     }
63 
64     // init n_rows and n_cols
65     n_rows = vv_strs.size();
66     n_cols = vv_strs[0].size();
67 
68     headers.resize(n_cols);
69 
70     // initializing headers
71     for(unsigned long long i_col{0}; i_col < n_cols; ++i_col)
72     {
73         if(is_first_col_header)
74         {
75             headers[i_col] = vv_strs[0][i_col];
76         }
77         else if(v_hdrs.size() > 0)
78         {
79             if(v_hdrs.size() == n_cols)
80             {
81                 headers[i_col] = v_hdrs[i_col];
82             }
83             else
84             {
85                 fmt::println(stderr, "Error: number of provided headers does not match with the number of
    columns in the data");
86                 exit(EXIT_FAILURE);
87             }
88         }
89         else
90         {
91             headers[i_col] = std::to_string(i_col + 1);
92         }
93     }
94 
95     for(unsigned long long i_col{0}; i_col < n_cols; ++i_col)
96     {
97         std::shorts::V_string values;
98         for(unsigned long long i_row{is_first_col_header}; i_row < n_rows; ++i_row)
99         {
```

```
100             if(vv_strs[i_row].size() != n_cols)
101             {
102                 fmt::println(stderr, "Error: inconsistent number of columns, check row {}", i_row + 1);
103                 exit(EXIT_FAILURE);
104             }
105 
106             values.emplace_back(vv_strs[i_row][i_col]);
107 
108             // check for the missig values
109             // missing values are empty string, NA, and NAN
110             if(vv_strs[i_row][i_col] == "" || vv_strs[i_row][i_col] == "NA" || vv_strs[i_row][i_col] ==
      "NAN")
111             {
112                 mising_values.insert({i_row, i_col});
113             }
114         }
115 
116         data[headers[i_col]] = values;
117     }
118 }
```

5.1.2.4 fill_data() [2/2]

```
void DF::DataFrame::fill_data (
            std::shorts::V_string const & v_strs,
            std::shorts::V_int const & v_cols_length,
            bool is_first_col_header = true,
            std::shorts::V_string v_hdrs = {} )  [private]
```

5.1.2.5 get_by_headers()

```
std::shorts::V_string DF::DataFrame::get_by_headers ( ) const
```

5.1.2.6 get_headers()

```
std::shorts::V_string DF::DataFrame::get_headers ( ) const
```

Get the headers.

Returns

std::shorts::V_string headers

5.1.2.7 get_n_cols()

```
int DF::DataFrame::get_n_cols ( ) const
```

Get the number cols of a given data.

Returns

int number of columns

Definition at line 10 of file ReadFiles.cpp.

```
11 {
12     return n_cols;
13 }
```

5.1.2.8 get_n_rows()

```
int DF::DataFrame::get_n_rows ( ) const
```

Get the number of rows of a given data.

Returns

int number of rows

Definition at line 15 of file ReadFiles.cpp.

```
16 {
17     return n_rows;
18 }
```

5.1.2.9 head()

```
void DF::DataFrame::head (
            unsigned long long n = 5 )
```

print n first rows off all columns

Parameters

n	

Definition at line 126 of file ReadFiles.cpp.

```
127 {
128     if(n > n_rows)
129     {
130         fmt::println("Warning: number of columns requested to be printed is bigger than number of
    available rows in data");
131         n = n_rows;
132     }
133 
134     for(auto const& curr_hdr : headers)
```

```
135     {
136         fmt::print("{:10} ", curr_hdr);
137     }
138
139     std::cout « '\n';
140
141     for(unsigned long long i{0}; i < n; ++i)
142     {
143         for(auto const& curr_hdr : headers)
144         {
145             fmt::print("{:10} ", data[curr_hdr][i]);
146         }
147
148         std::cout « "\n";
149     }
150 }
```

5.1.2.10 read_files() [1/2]

```
void DF::DataFrame::read_files (
              std::string_view path,
              char delim = ',',
              bool is_first_col_header = true,
              std::shorts::V_string v_hdrs = {} )
```

read files

Parameters

path	path to input file
delim	delimiter for parsing the input file
is_first_col_header	

Definition at line 120 of file ReadFiles.cpp.

```
121 {
122     auto v_strs = read_lines(path);
123     fill_data(v_strs, delim, is_first_col_header, v_hdrs);
124 }
```

5.1.2.11 read_files() [2/2]

```
void DF::DataFrame::read_files (
              std::string_view path,
              std::shorts::V_int const & v_cols_length,
              bool is_first_col_header = true,
              std::shorts::V_string v_hdrs = {} )
```

Parameters

path	
v_cols_length	
is_first_col_header	

5.1.2.12 read_lines()

```
std::shorts::V_string DF::DataFrame::read_lines (
            std::string_view path )  [private]
```

Definition at line 20 of file ReadFiles.cpp.

```
21 {
22     std::ifstream ifs(path.data());
23 
24     if(ifs.fail())
25     {
26         fmt::println("Error: unable to read file {}.\nPlease check your input.", path);
27         exit(EXIT_FAILURE);
28     }
29 
30 
31     std::string line;
32     std::shorts::V_string v_strs;
33 
34 
35     while(std::getline(ifs, line))
36     {
37         if(line.size() == 0) continue;
38         line.erase(std::remove(line.begin(), line.end(), '\r'),line.end());
39         v_strs.emplace_back(line);
40     }
41 
42     return v_strs;
43 }
```

5.1.2.13 remove_duplications()

```
void DF::DataFrame::remove_duplications (
            std::string const & hdr )
```

get a given header and only keep the first occurance and remove the remaning rows

Parameters

hdr	header to check the duplication

5.1.2.14 set_headers()

```
void DF::DataFrame::set_headers (
            std::shorts::V_string const & v_hdrs )
```

Set the headers with user provided vector of strings.

Parameters

v_hdrs	provided headers from users

5.1.3 Member Data Documentation

5.1.3.1 data

`std::shorts::Data DF::DataFrame::data`

saving data in an unrodered map

Definition at line 54 of file ReadFiles.hpp.

5.1.3.2 headers

`std::shorts::V_string DF::DataFrame::headers`

Header of data, whether is the first line or will be provided by user in case of user not providing header it would be string number start from one to number of columns of the data.

Definition at line 73 of file ReadFiles.hpp.

5.1.3.3 mising_values

`std::unordered_map<int, int> DF::DataFrame::mising_values`

an unorderd maps for missing values

Definition at line 79 of file ReadFiles.hpp.

5.1.3.4 n_cols

`unsigned long long DF::DataFrame::n_cols`

number of columns

Definition at line 66 of file ReadFiles.hpp.

5.1.3.5 n_rows

`unsigned long long DF::DataFrame::n_rows`

number of rows

Definition at line 60 of file ReadFiles.hpp.

The documentation for this class was generated from the following files:

- include/ReadFiles.hpp
- src/ReadFiles.cpp

Chapter 6

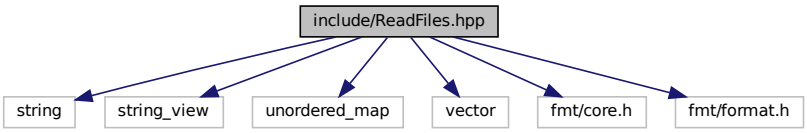
File Documentation

6.1 include/ReadFiles.hpp File Reference

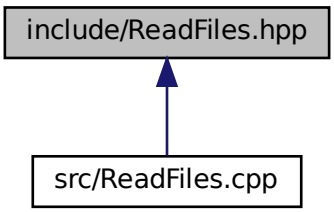
A class for reading files with different delimiters.

```
#include <string>
#include <string_view>
#include <unordered_map>
#include <vector>
#include "fmt/core.h"
#include "fmt/format.h"
```

Include dependency graph for ReadFiles.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class DF::DataFrame

  DataFrame is class for parsing data in a given file with a give delimeter (default is comma ','). all data will be saved as string wich user can later covert to desired type.

Namespaces

- std
- std::shorts

  namespace for introducting shortnames
- DF

Typedefs

- using std::shorts::V_string = vector< string >
- using std::shorts::VV_string = vector< V_string >
- using std::shorts::Data = unordered_map< string, V_string >
- using std::shorts::V_double = vector< double >
- using std::shorts::V_int = vector< int >

6.1.1 Detailed Description

A class for reading files with different delimiters.

Author

Naeim Moafinejad (snmoafinejad@iimcb.gov.pl, s.naeim.moafi.n@gmail.com)

Version

0.1

Date

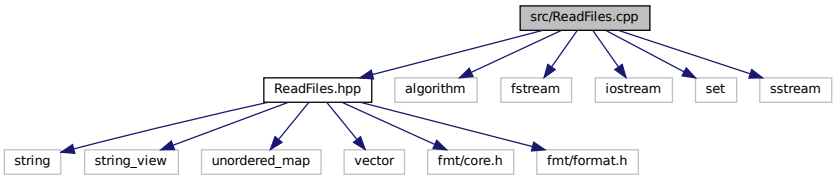
2024-10-29

Copyright

6.2 src/ReadFiles.cpp File Reference

```
#include "ReadFiles.hpp"
#include <algorithm>
#include <fstream>
#include <iostream>
#include <set>
#include <sstream>
```

Include dependency graph for ReadFiles.cpp:



Functions

- int main ()

6.2.1 Function Documentation

6.2.1.1 main()

```
int main ( )
```

Definition at line 153 of file ReadFiles.cpp.

```
154 {
155     DF::DataFrame df;
156 
157 
158     df.read_files("test.txt", '\t');
159     df.head();
160     std::cout << '\n';
161 
162     df.read_files("test.txt", '\t', false);
163     df.head();
164     std::cout << '\n';
165 
166     df.read_files("test.txt", '\t', false, {"1.0000", "2.0000", "3.0000", "4.0000", "5.0000"});
167     df.head();
168     std::cout << '\n';
169 
170     return EXIT_SUCCESS;
171 }
```

Index