End-user documentation

May 2022

1 Introduction

This is an end-user guidance document. It is intended to be a guide to provide a detailed description of the *ReadCSVFile* class, its member functions, its inputs and outputs, and its dependencies. Including, a detailed description of the *DisplayDoubleVector* function and its input parameters and returns. Both the class and function are implemented in C++, which belongs to the library *ReadCSVFile* and its associated *ReadCSVFile.h* and *ReadCSVFile.cpp* files.

2 Class: ReadCSVFile

2.1 Summary

A class to read a CSV (comma-separated values) file and store the content into a double vector of string, i.e. vector<vector<string>>, named *datafrme* in this implementation.

The constructor receives a filename as string data type then read the CSV file according to the given name which is in the same directory as the program. And the object will stores the content in a double vector of string, which its rows and columns represent the rows and columns of the file.

The class contains functions for returning the content in the CSV file such as a function to return the variable double vector of string, vector<vector<string>>, a fucntion for returning only the values in the CSV file, i.e. the file header (assumed to be the 1st row) and index (assumed to be the 1st column) will be omitted, and a function to return an associated value according to the input parameters locate the position in the datafrme.

2.1.1 Dependencies

This implemented class uses some standard C++ library which are *iostream*, *string*, *vector*, *fstream* and *sstream*.

2.1.2 Inputs

This class requires only one input filename (string data type) to construct an object.

2.1.3 Functions

get

a get function to return the dataframe (vector<vector<string>> data type). See exmaple 2.4.1.

get_values

a get function to return only the values (vector<vector<double>> data type) in the dataframe, the header (1st row) and the index (1st column) will be removed. See exmaple 2.4.2.

get_value

a get function requires users input a position cell in the datafrme to return a single value (double data type) associated with the input parameters locate the position in the datafrme. See exmaple 2.4.3.

2.2 Input Parameters

filename

string data type: The CSV file name that belongs to the same directory as the program.

2.3 Functions

2.3.1 Constructor

A constructor: ReadCSVFile(string filename)
receives a filename as string data type then read the CSV file according to the given name which belongs to the same directory as the program, then stores the content in a double vector of string named dataframe.
Parameters:
- string data type: filename

2.3.2 get

A public function: get()
a get function to return the dataframe.
Parameters:
- No input parameters required for this function.
Returns:

- vector<vector<string>> data type: dataframe

2.3.3 get_values

A public function: get_values()

a get function to return only values (type double) of the dataframe, the first row and column will be removed.

Parameters:

- No input parameters required for this function.

Returns:

- vector<vector<double>> data type: values in every cells of the dataframe without any header row and index column.

2.3.4 get_value

A public function: get_value(string row_name, string col_name)

a get function to return a single value that matches the given row name (row_name) and column name (col_name). If no cells in the dataframe are found matching the given name or data, then display an error and return 0.0.

Parameters:

- string data type: row_name used to locate the cell in row that its name associated with this string.

- string data type: col_name used to locate the cell in column that its name associated with this string.

Returns:

- double data type: a single value in the associated cell.

2.4 Example

2.4.1 example 1

```
#include <string>
#include <vector>
#include "ReadCSVFile.h"

int main(){
    string filename = "StockDataMar2022.csv";
    ReadCSVFile df(filename); // read csv file
    vector<vector<string>> sdf = df.get();
    DisplayDoubleVector(sdf); // display the content
    return 0;
}
```

output

```
STOCK PRICES MAR 2022 Price R sigma
Bentley 180 0.04 0.2
Tesla 120 0.1 0.15
```

```
Ford 150 0.05 0.17
```

2.4.2 example 2

```
#include <string>
#include <vector>
#include "ReadCSVFile.h"

int main(){
    string filename = "StockDataMar2022.csv";
    ReadCSVFile df(filename); // read csv file
    vector<vector<double>> ddf = df.get_values();
    DisplayDoubleVector(ddf); // display the content
    return 0;
}
```

output

```
180 0.04 0.2
120 0.1 0.15
150 0.05 0.17
```

2.4.3 example 3

```
#include <string>
#include <iostream>
#include "ReadCSVFile.h"

using namespace std;

int main(){
    string filename = "StockDataMar2022.csv";
    ReadCSVFile df(filename); // read csv file
    //return a value in the dataframe
    //corresponds to a given stock name and column data
    double S0 = df.get_value("Tesla", "Price");
    double R = df.get_value("Tesla", "R");
    double sigma = df.get_value("Tesla", "sigma");

    cout << "Tesla stock option::"
         << " S0=" << S0
         << ", r=" << R
         << ", sigma=" << sigma
         << endl;
    return 0;
}
```

output

```
Tesla stock option:: S0=120, r=0.1, sigma=0.15
```

3 Function: DisplayDoubleVector

A template function to display the contents of any double vector by passing double vector of any data type and outputting its contents through the terminal prompt.

3.1 Dependencies

This implemented function uses some standard C++ library, which are *iostream* and *vector*.

3.2 Inputs

This function requires a user to pass a double vector of any data type, e.g. *double* or *string*. (see example 3.4.1,3.4.2,3.4.3,3.4.4)

3.3 Output

This function outputs the contents in the given double vector through the terminal prompt, displaying each vector line by line and separating each the vector member with spaces.

3.4 Example

3.4.1 example 1

```
#include <vector>
#include "ReadCSVFile.h"

int main(){
    vector<vector<int>> ivec = {{1,2,3}, {2,4,8}};
    DisplayDoubleVector(ivec);

    return 0;
}
```

output

```
1 2 3
2 4 8
```

3.4.2 example 2

```
#include <vector>
#include "ReadCSVFile.h"

int main(){
    vector<vector<double>> dvec = {{1.0,2.0,3.0}, {2.4,4.4,8.8}};
    DisplayDoubleVector(dvec);

    return 0;
}
```

output

```
1 2 3
2.4 4.4 8.8
```

3.4.3 example 3

```
#include <vector>
#include "ReadCSVFile.h"

int main(){
    vector<vector<char>> cvec = {{'C', '+', '+'},
                                 {'v', 'i', 'r'}};
    DisplayDoubleVector(cvec);

    return 0;
}
```

output

```
C++
v i r
```

3.4.4 example 4

```
#include <vector>
#include <string>
#include "ReadCSVFile.h"

int main(){
    vector<vector<string>> svec = {{"1st", "2nd", "3rd"},
                                   {"val", "in", "row"}};
    DisplayDoubleVector(svec);

    return 0;
```

```
}
```

output

```
1st 2nd 3rd
val in row
```