
Table of Contents

EXTERNAL FUNCTION TO READ .csv FILES AND CONVERT TO .mat	1
Read the file	2
Take the data, number of rows and number of columns	2
Is There a header in the file?	2
Dictionary creation	2

EXTERNAL FUNCTION TO READ .csv FILES AND CONVERT TO .mat

```
function [data,column_names,string_conversions] =  
    string_CSV_read(filename)  
  
% string_CSV_read This function reads the data contain in a .csv file  
% and  
% transform it for its use. The function returns the next values:  
  
% data: contains the readed data. The columns that contains nominal  
% values are transformed into numerical ones.  
% column_names: if the file contains a header, this variable saves  
% the name of each column(*).  
% string_conversions: this variable contains the nominal values of  
% the columns that have been transformed. If the column has  
% numerical value, it contains {}. Otherwise, the nominal values  
% are saved in column position.  
% (*)NOTE: In case of a entire nominal dataset with no header, the  
first  
% example can be confused with header.  
%  
%  
% Example: the file contains the following data:  
%      A,B,C,D  
%      1,2,Sun,YES  
%      3,1,Rain,YES  
%      3,5,Sun,NO  
%  
% Then, the function returns:  
%      - data = [ 1 2 2 2  
%                3 1 1 2  
%                3 5 2 1]  
%      - column_names = {'A','B','C','D'}  
%      - string_conversions = {{}  
%                               {}  
%                               {'Rain' 'Sun'}  
%                               {'NO' 'YES'}}  
  
% Created by:  
% Pedro L#pez Garc#a (Phd. Student, University of Deusto,Bilbao)
```

```
% Enrique Onieva Caracuel (PostDoctoral Researcher & Project Manager,  
University of Deusto, Bilbao).  
% Twitter: @EnriqueOnieva  
% Research Gate: https://www.researchgate.net/profile/Enrique_Onieva  
% https://www.researchgate.net/profile/  
Pedro_Lopez_Garcia
```

```
fid = fopen(filename,'r');  
filestrings = [];
```

Not enough input arguments.

Error in string_CSV_read (line 43)
fid = fopen(filename,'r');

Read the file

```
while 1  
    line = fgetl(fid);  
    if ~ischar(line),break,end  
    tmp = regexp(line,'([^\s:]*)','tokens');  
    str = cat(2,tmp{:});  
    filestrings = cat(1,filestrings,str);  
end
```

Take the data, number of rows and number of columns

```
data = str2double(filestrings);  
nrows = size(data,1);  
ncolumns = size(data,2);
```

Is There a header in the file?

```
column_names = {};  
if isnan(data(1,:)) == ones(1, ncolumns)  
    column_names = filestrings(1,:);  
    data = data(2:end, :);  
nrows=nrows-1;  
end
```

Dictionary creation

```
string_conversions = {};  
for i = 1: ncolumns  
    % if the columns contains nominal values  
    if isnan(data(:,i)) == ones (nrows, 1)  
        %Take these values and save it in string_conversions  
        names = unique(filestrings(2:end,i));  
        string_conversions{i} = names;
```

```
    % Transform the names into numerical value
    for j = 1: numel(names)
        index = find(strcmp(filestrings(:,i), names{j}));
        for k = 1: size(index, 1)
            data(index(k)-1, i) = j;
        end
    end
else
    string_conversions{i} = {};
end
end
```

Published with MATLAB® R2015b