EGR115

Using `getcsv()`

From the help file included with the `getcsv()` and `putcsv()` .p files:

```
[num, txt, raw] = getcsv(fname)

 Reads the file "fname" and parses it into numeric,
 text, and combined data.  Each array is the same
 size.  Elements of the CSV file which are not numeric
 are replaced in the numeric array with NaN.  Elements
 of the CSV file which are numeric or empty are replaced
 in the text array with the empty string, ''.

 The combined "raw" array is a merge of the numeric and
 text data with numeric values taking precedence over '', and
 '' taking precedence over NaN

File "fname" must be available for writing in order to correct
for a flaw in Excel that writes out CSV files incorrectly. Be
sure the file is not open in Excel when calling getcsv().


 Parameters:
 fname - a string for a CSV file to be read

 Return values:
   If fname can be read:
       num - numeric vector or matrix of numeric data in file
       txt - cell array of non-numeric data in file
       raw - cell array of combined numeric and string data
   Otherwise:
       num = false
       txt = false
       raw = false
```

Summary:

- `getcsv()` requires a file name string as argument. This file name must include the `.csv` extension.
- The file to be read should be in the same folder as the program, and must be available for writing (meaning it cannot be open in Excel, LibreOffice, or other locking programs).
- `getcsv()` returns three arrays: the first is numeric (containing just the numeric data in the file) and the final two are cell arrays: the second returned array contains just the non-numeric data in the file; and the third returned array contains all data from the file.
- Always test for successful reading before trying to use the file data in your program: check the return values! If any are Boolean values, the reading was unsuccesful. This is typically implemented as:

```
[num, txt, raw] = getcsv('my_file.csv');

if islogical(raw)
   fprintf('Error reading file!\n');

else
   % Successfully read the file.
   …

end
```