csv.module

`csv.module`

Reads and writes CSV files

Requires str.lg, rs.lg and df.lg

csv.read

`csv.read :in [types []] [sep ",] [dec ".] [header "true] [skip 0]`

Reads a csv file in accordance with RFC 4180, with some differences detailed in the notes

Parameters

in: CSV file name

types: list of variables (columns) data types ("char" or "num"), default []

sep: field separator character, default ","

dec: separator character for decimal numbers, default "."

header: is there a header in the CSV file? default "true"

skip: the number of lines of the CSV file to skip before beginning to read data, default 0

Returns

A list of lists. When data types are not defined each sublist represents a row in the CSV file.

Otherwise returns a dataframe, defined as a list composed of two lists: a first list of data types ("char" for character variables, "num" for numeric variables) and a second one composed of sentences (lists of words) representing the variables in the dataframe, where in each sentence the first word is the variable name and the others its values. If types are defined, non numeric values in a numeric columnn in the CSV file are set to missing as empty fields in a numeric or character column.

Notes

In UCBLogo, words (they would be called a character string in other languages) are one of the three kinds of information (arrays and lists the other two) that Logo can process (numbers are a special case of words). A list of words, such as [1 2 3], is called a sentence or flat list.

1. csv.read adds a header line if not present in the CSV file.
2. Line breaks may be LF (char 10), CR (char 13) or CRLF.
3. A field may contain different data types in different lines. If types are defined, non numeric values in a numeric columnn in the CSV file are set to missing as empty fields in a numeric or character column.
4. UTF-8 BOM Byte Order Mark (Decimal representation : 239 187 191) detection and removal. The function gives a warning.
5. If in the csv file are used special UCBlogo characters (vertical bar, semicolon, ...) as field separators, in csv.read they should be backslashed.

6. Numbers in UCBLogo are integers (1,2, ...) and decimals (1.1, ...). Specifying "dec" as "," allows to convert comma separated numbers (3,14) in the form recognized by UCBLogo.
7. No LF or CR characters in dataframe's variables

Examples

```
make "out csv.transpose (list se "id iseq 1 4 se "sex (rs.rep [female male] 2) se "age rs.rand 1 100 4)
csv.write  :out "list.csv
show csv.read "list.csv
show (csv.read "list.csv [num char num])
make "out csv.transpose (list se "id iseq 1 4 se "sex [female male || male] se "age [27 aabb 32 ||])
csv.write  :out "list.csv
show csv.read "list.csv
show (csv.read "list.csv [num char num])
```

csv.write

`csv.write :list :out [sep ",]`

Writes a csv file from a list of lists. When data types are not defined each sublist represents a row in the CSV file, otherwise the list of list is a two elements list: a list of data types and a list composed of sublists representing the columns in the CSV file.

Parameters

list: list of lists

out: CSV file name

sep: separator character, default ","

Notes

1. csv.write quotes field values containg the separator character and/or "
2. " in a field is quoted adding another "
3. By default fields are separated by ","

Examples

```
make "list csv.transpose (list se "id iseq 1 20 se "sex (rs.rep [female male] 10) se "age rs.rand 1 100 20)
csv.write  :list "list.csv
make "df list [num char num] csv.transpose :list
csv.write  :df "df.csv
```

csv.read.lines

`csv.read.lines :in  [sep ",]`

Reads a simple csv file -- an header should be present, no newline characters (CR, LF) in the fields, separator character for decimal numbers should be "." -- line by line. Empty fields are set to missing.

Parameters

in: CSV file name

sep: field separator character, default ","

Returns

A list of lists, each sublist represents a row in the CSV file.

Notes

1. Line breaks may be LF (char 10), CR (char 13) or CRLF.
2. A field may contain different data types in different lines.
3. By default fields are separated by commas. They may be separated by another character.
4. UTF-8 BOM Byte Order Mark (Decimal representation : 239 187 191) detection and removal.
5. If in the csv file are used special UCBlogo characters (vertical bar, semicolon, ...) as field separators, in csv.read.lines they should be backslashed.

Examples

```
make "out csv.transpose (list se "id iseq 1 20 se "sex (rs.rep [female male] 10) se "age rs.rand 1 100 20)

csv.write  :out "list.csv

make "list csv.read.lines "list.csv
```