Package: easycsv (via r-universe)

September 8, 2024

Type Package

Title Load Multiple 'csv' and 'txt' Tables

Date 2018-04-27

Version 1.0.8

Description Allows users to easily read multiple comma separated tables and create a data frame under the same name. Is able to read multiple comma separated tables from a local directory, a zip file or a zip file on a remote directory.

Depends R (>= 3.2.3)

Imports data.table (>= 1.10)

License GPL-2

URL https://github.com/bogind/easycsv

BugReports https://github.com/bogind/easycsv/issues

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Repository https://bogind.r-universe.dev

RemoteUrl https://github.com/bogind/easycsv

RemoteRef HEAD

RemoteSha 168965d1042846ea92284083597daa467c863a3c

Contents

choose_dir *Choose a Folder Interactively*

Description

Use a folder widget to choose a folder interactively.

Usage

```
choose_dir()
```

Details

This brings up folder selection widget, it requires no arguments and is implemented into fread_folder & loadcsv_multi as the default if no directory is supplied. Currently works only on mac OS,windows and Linux. for the windows implementation and further detail see choose.dir(remote url).

Value

A length-one character vector, character NA if 'Cancel' was selected.

See Also

choose.dir(remote url), Identify.OS

fread_folder *read multiple csv files into named data frames*

Description

Reads multiple files in table format using fread's speed and creates a data frame from them, with cases corresponding to lines and variables to fields in the file.

Usage

```
fread_folder(directory = NULL,
                          extension = "CSV",
                          sep = "auto",
                          nrows = -1L,
                          header = "auto",
                          na.strings = "NA",
                          stringsAsFactors = FALSE,
                          verbose=getOption("datatable.verbose"),
                          skip = 0L,
                          drop = NULL,
                          colClasses = NULL,
```

```
integer64=getOption("datatable.integer64"),# default:"integer64"
        dec = if (sep!=".") "." else ",",
        check.names = FALSE,
        encoding = "unknown",
        quote = "\"",
        strip.white = TRUE,
        fill = FALSE,
        blank.lines.skip = FALSE,
        key = NULL,
        Names=NULL,
        prefix=NULL,
        showProgress = interactive(),
        data.table=TRUE
        )
```

Arguments

directory	a directory to load the files from, if NULL then a manual choice is provided on windows OS.
extension	"TXT" for tables in '.txt' files, "CSV" for tables in '.csv' files, "BOTH" for both file endings.
sep	The separator between columns. Defaults to the first character in the set [,\t |;:] that exists on line autostart outside quoted ("") regions, and separates the rows above autostart into a consistent number of fields, too.
nrows	The number of rows to read, by default -1 means all. Unlike read.table, it doesn't help speed to set this to the number of rows in the file (or an estimate), since the number of rows is automatically determined and is already fast. Only set nrows if you require the first 10 rows, for example. 'nrows=0' is a special case that just returns the column names and types; e.g., a dry run for a large file or to quickly check format consistency of a set of files before starting to read any.
header	Does the first data line contain column names? Defaults according to whether every non-empty field on the first data line is type character. If so, or TRUE is supplied, any empty column names are given a default name.
na.strings	A character vector of strings which are to be interpreted as NA values. By default ",," for columns read as type character is read as a blank string ("") and ",NA," is read as NA. Typical alternatives might be na.strings=NULL (no coercion to NA at all!) or perhaps na.strings=c("NA","N/A","null")
stringsAsFactors	Convert all character columns to factors?
verbose	Be chatty and report timings?
skip	If 0 (default) use the procedure described below starting on line autostart to find the first data row. skip>0 means ignore autostart and take line skip+1 as the first data row (or column names according to header="auto"|TRUE|FALSE as usual). skip="string" searches for "string" in the file (e.g. a substring of the column names row) and starts on that line (inspired by read.xls in package gdata).
drop	Vector of column names or numbers to drop, keep the rest.

`colClasses` A character vector of classes (named or unnamed), as read.csv. Or a named list of vectors of column names or numbers, see examples. colClasses in fread is intended for rare overrides, not for routine use. fread will only promote a column to a higher type if colClasses requests it. It won't downgrade a column to a lower type since NAs would result. You have to coerce such columns afterwards yourself, if you really require data loss.

`integer64` "integer64" (default) reads columns detected as containing integers larger than 2^31 as type bit64::integer64. Alternatively, "double"|"numeric" reads as base::read.csv does; i.e., possibly with loss of precision and if so silently. Or, "character".

`dec` The decimal separator as in base::read.csv. If not "." (default) then usually ",". See details.

`check.names` default is FALSE. If TRUE then the names of the variables in the data.table are checked to ensure that they are syntactically valid variable names. If necessary they are adjusted (by make.names) so that they are, and also to ensure that there are no duplicates.

`encoding` default is "unknown". Other possible options are "UTF-8" and "Latin-1". Note: it is not used to re-encode the input, rather enables handling of encoded strings in their native encoding.

`quote` By default ("\""), if a field starts with a doublequote, fread handles embedded quotes robustly as explained under Details. If it fails, then another attempt is made to read the field as is, i.e., as if quotes are disabled. By setting quote="", the field is always read as if quotes are disabled.

`strip.white` default is TRUE. Strips leading and trailing whitespaces of unquoted fields. If FALSE, only header trailing spaces are removed.

`fill` logical (default is FALSE). If TRUE then in case the rows have unequal length, blank fields are implicitly filled.

`blank.lines.skip` logical, default is FALSE. If TRUE blank lines in the input are ignored.

`key` Character vector of one or more column names which is passed to setkey. It may be a single comma separated string such as key="x,y,z", or a vector of names such as key=c("x","y","z"). Only valid when argument data.table=TRUE

`Names` A character vector of names for the tables to be read, note that the table will be read and listed by an alphabetical order, use with caution.

`prefix` A character string to be prefixed to each table name.

`showProgress` TRUE displays progress on the console using \r. It is produced in fread's C code where the very nice (but R level) txtProgressBar and tkProgressBar are not easily available.

`data.table` logical. TRUE returns a data.table. FALSE returns a data.frame.

Details

Similar to loadcsv_multi can read multiple tables from either '.txt' or '.csv' files, uses fread for additional speed. Takes arguments that respond to fread's arguments.

Value

A data.frame containing a representation of the data in the file.

Note

This function alone requires fread, it is not installed by default with easycsv, because of that. If you use "BOTH" option with 'txt' make sure your '.txt' and '.csv' files have different names.

See Also

loadZIPcsvfromURL loadcsvfromZIP loadcsv_multi fread

Examples

```
require(easycsv)
require("data.table")
directory = getwd()
write.csv(data.frame(matrix(1:9, nrow = 3)), file = file.path(directory,"/table1.csv"))
write.csv(data.frame(matrix(1:9, nrow = 3)), file = file.path(directory,"/table2.csv"))
write.csv(data.frame(matrix(1:9, nrow = 3)), file = file.path(directory,"/table3.txt"))
write.csv(data.frame(matrix(1:9, nrow = 3)), file = file.path(directory,"/table4.txt"))
fread_folder(directory, extension = "BOTH")
```

---

fread_zip *read multiple csv files into named data frames*

Description

Reads multiple files in table format using fread's speed and creates a data frame from them, with cases corresponding to lines and variables to fields in the file. works on .zip files only.

Usage

```
fread_zip(filezip = NULL,
                        extension = "BOTH",
                        sep = "auto",
                        nrows = -1L,
                        header = "auto",
                        na.strings = "NA",
                        stringsAsFactors = FALSE,
                        verbose=getOption("datatable.verbose"),
                        autostart = 1L,
                        skip = 0L,
                        drop = NULL,
                        colClasses = NULL,
                  integer64=getOption("datatable.integer64"),# default:"integer64"
                        dec = if (sep!=".") "." else ",",
                        check.names = FALSE,
```

```
                    encoding = "unknown",
                    quote = "\"",
                    strip.white = TRUE,
                    fill = FALSE,
                    blank.lines.skip = FALSE,
                    key = NULL,
                    Names=NULL,
                    prefix=NULL,
                    showProgress = interactive(),   # default: TRUE
                    data.table=TRUE
                      )
```

Arguments

filezip	a '.zip' file to load the files from, if NULL then a manual choice is provided. does not work with '.rar' files
extension	"TXT" for tables in '.txt' files, "CSV" for tables in '.csv' files, "BOTH" for both file endings.
sep	The separator between columns. Defaults to the first character in the set [,\t |;:] that exists on line autostart outside quoted ("") regions, and separates the rows above autostart into a consistent number of fields, too.
nrows	The number of rows to read, by default -1 means all. Unlike read.table, it doesn't help speed to set this to the number of rows in the file (or an estimate), since the number of rows is automatically determined and is already fast. Only set nrows if you require the first 10 rows, for example. 'nrows=0' is a special case that just returns the column names and types; e.g., a dry run for a large file or to quickly check format consistency of a set of files before starting to read any.
header	Does the first data line contain column names? Defaults according to whether every non-empty field on the first data line is type character. If so, or TRUE is supplied, any empty column names are given a default name.
na.strings	A character vector of strings which are to be interpreted as NA values. By default ",," for columns read as type character is read as a blank string ("") and ",NA," is read as NA. Typical alternatives might be na.strings=NULL (no coercion to NA at all!) or perhaps na.strings=c("NA","N/A","null")
stringsAsFactors	Convert all character columns to factors?
verbose	Be chatty and report timings?
autostart	Any line number within the region of machine readable delimited text, by default 30. If the file is shorter or this line is empty (e.g. short files with trailing blank lines) then the last non empty line (with a non empty line above that) is used. This line and the lines above it are used to auto detect sep and the number of fields. It's extremely unlikely that autostart should ever need to be changed, we hope.
skip	If 0 (default) use the procedure described below starting on line autostart to find the first data row. skip>0 means ignore autostart and take line skip+1 as the first data row (or column names according to header="auto"|TRUE|FALSE as usual).

skip="string" searches for "string" in the file (e.g. a substring of the column names row) and starts on that line (inspired by read.xls in package gdata).

`drop` Vector of column names or numbers to drop, keep the rest.

`colClasses` A character vector of classes (named or unnamed), as read.csv. Or a named list of vectors of column names or numbers, see examples. colClasses in fread is intended for rare overrides, not for routine use. fread will only promote a column to a higher type if colClasses requests it. It won't downgrade a column to a lower type since NAs would result. You have to coerce such columns afterwards yourself, if you really require data loss.

`integer64` "integer64" (default) reads columns detected as containing integers larger than 2^31 as type bit64::integer64. Alternatively, "double"|"numeric" reads as base::read.csv does; i.e., possibly with loss of precision and if so silently. Or, "character".

`dec` The decimal separator as in base::read.csv. If not "." (default) then usually ",". See details.

`check.names` default is FALSE. If TRUE then the names of the variables in the data.table are checked to ensure that they are syntactically valid variable names. If necessary they are adjusted (by make.names) so that they are, and also to ensure that there are no duplicates.

`encoding` default is "unknown". Other possible options are "UTF-8" and "Latin-1". Note: it is not used to re-encode the input, rather enables handling of encoded strings in their native encoding.

`quote` By default ("\""), if a field starts with a doublequote, fread handles embedded quotes robustly as explained under Details. If it fails, then another attempt is made to read the field as is, i.e., as if quotes are disabled. By setting quote="", the field is always read as if quotes are disabled.

`strip.white` default is TRUE. Strips leading and trailing whitespaces of unquoted fields. If FALSE, only header trailing spaces are removed.

`fill` logical (default is FALSE). If TRUE then in case the rows have unequal length, blank fields are implicitly filled.

`blank.lines.skip` logical, default is FALSE. If TRUE blank lines in the input are ignored.

`key` Character vector of one or more column names which is passed to setkey. It may be a single comma separated string such as key="x,y,z", or a vector of names such as key=c("x","y","z"). Only valid when argument data.table=TRUE

`Names` A character vector of names for the tables to be read, note that the table will be read and listed by an alphabetical order, use with caution.

`prefix` A character string to be prefixed to each table name.

`showProgress` TRUE displays progress on the console using `\r`. It is produced in fread's C code where the very nice (but R level) txtProgressBar and tkProgressBar are not easily available.

`data.table` logical. TRUE returns a data.table. FALSE returns a data.frame.

Details

Similar to loadcsv_multi can read multiple tables from either '.txt' or '.csv' files, uses fread for additional speed. Takes arguments that respond to fread's arguments.

Value

A data.frame containing a representation of the data in the file.

Note

This function alone requires fread, it is not installed by default with easycsv, because of that. If you use "BOTH" option with 'txt' make sure your '.txt' and '.csv' files have different names.

See Also

loadZIPcsvfromURL loadcsvfromZIP loadcsv_multi fread_folder fread

Examples

```
require(easycsv)
filezip <- system.file("exampleZips", "example_tables.zip", package="easycsv")
fread_zip(filezip)
fread_zip(filezip, extension = "CSV")
```

Identify.OS *Returns Operating System*

Description

returns one obejct which identifies if the OS is supported for use of choose_dir.

Usage

```
Identify.OS()
```

Details

Internal function of choose_dir, can be used to identify only one of its' supported operating systems.

Value

A character object.

See Also

choose_dir, .Platform, Sys.info

`loadcsvfromZIP` *read multiple csv files into named data frames*

Description

Reads multiple files in table format and creates a data frame from them, with cases corresponding to lines and variables to fields in the file.

Usage

```
loadcsvfromZIP(filezip = NULL,
                            txt = FALSE ,
                            encoding = "Latin-1",
                            stringsAsFactors = FALSE,
                            header = TRUE,
                            quote = "\"",
                            fill = TRUE,
                            comment.char = "")
```

Arguments

`filezip`	a '.zip' file to load the files from, if NULL then a manual choice is provided. does not work with '.rar' files
`txt`	logical. if TRUE .txt files will be loaded as tables instead of .csv
`encoding`	character. files encoding. default is Latin-1
`stringsAsFactors`	logical: should character vectors be converted to factors? Note that this is overridden by as.is and colClasses, both of which allow finer control.
`header`	a logical value indicating whether the files contain the names of the variables as its first line. If missing, the value is determined from the file format: header is set to TRUE if and only if the first row contains one fewer field than the number of columns.
`quote`	the set of quoting characters. To disable quoting altogether, use quote = "". See scan for the behaviour on quotes embedded in quotes. Quoting is only considered for columns read as character, which is all of them unless colClasses is specified.
`fill`	logical. If TRUE then in case the rows have unequal length, blank fields are implicitly added.
`comment.char`	character: a character vector of length one containing a single character or an empty string. Use "" to turn off the interpretation of comments altogether.

Details

loadcsv_multi is used for uncompressed files in a single folder.it can be used either by entering the local directory the files are in, or just running it with no arguments for manual folder selection on windows OS. It receives some arguments from read.csv and they are listed in the arguments section. loadcsvfromZIP is used for comma separated tables inside of a .zip file. loadZIPcsvfromURL is used for comma separated tables inside of a .zip file on the internet, no download needed.

Value

A data.frame containing a representation of the data in the file.

See Also

loadZIPcsvfromURL loadcsv_multi

Examples

```
require(easycsv)
filezip <- system.file("exampleZips", "example_tables.zip", package="easycsv")
loadcsvfromZIP(filezip)
loadcsvfromZIP(filezip, txt = TRUE)
```

`loadcsv_multi` *read multiple csv files into named data frames*

Description

Reads multiple files in table format and creates a data frame from them, with cases corresponding to lines and variables to fields in the file.

Usage

```
loadcsv_multi(directory = NULL,
                          extension = "CSV",
                          encoding = "Latin-1",
                          stringsAsFactors = FALSE,
                          header = TRUE,
                          quote = "\"",
                          fill = TRUE,
                          comment.char = "")
```

Arguments

directory	a directory to load the files from, if NULL then a manual choice is provided on windows OS.
extension	logical. if TRUE .txt files will be loaded as tables instead of .csv.
encoding	character. files encoding. default is Latin-1
stringsAsFactors	logical: should character vectors be converted to factors? Note that this is overridden by as.is and colClasses, both of which allow finer control.
header	a logical value indicating whether the files contain the names of the variables as its first line. If missing, the value is determined from the file format: header is set to TRUE if and only if the first row contains one fewer field than the number of columns.
quote	the set of quoting characters. To disable quoting altogether, use quote = "". See scan for the behavior on quotes embedded in quotes. Quoting is only considered for columns read as character, which is all of them unless colClasses is specified.
fill	logical. If TRUE then in case the rows have unequal length, blank fields are implicitly added.
comment.char	character: a character vector of length one containing a single character or an empty string. Use "" to turn off the interpretation of comments altogether.

Details

loadcsv_multi is used for uncompressed files in a single folder.it can be used either by entering the local directory the files are in, or just running it with no arguments for manual folder selection on windows OS. It receives some arguments from read.csv and they are listed in the arguments section. loadcsvfromZIP is used for comma separated tables inside of a .zip file. loadZIPcsvfromURL is used for comma separated tables inside of a .zip file on the internet, no download needed.

Value

A data.frame containing a representation of the data in the file.

See Also

loadZIPcsvfromURL loadcsvfromZIP

Examples

```
require(easycsv)
directory = getwd()
table1 <- data.frame(matrix(1:9, nrow = 3))
write.csv(table1, file = file.path(directory,"/table1.csv"))
write.csv(table1, file = file.path(directory,"/table2.txt"))
loadcsv_multi(directory, extension = "BOTH")
```

`loadZIPcsvfromURL` *read multiple csv files into named data frames*

Description

Reads multiple files in table format and creates a data frame from them, with cases corresponding to lines and variables to fields in the file.

Usage

```
loadZIPcsvfromURL(urlAddress = NULL,
                              txt = FALSE ,
                            encoding = "Latin-1",
                            stringsAsFactors = FALSE,
                            header = TRUE,
                            quote = "\"",
                            fill = TRUE,
                            comment.char = "")
```

Arguments

`urlAddress`	a URL address of the zipped file, if NULL then Israeli GTFS will be downloaded and added to env.
`txt`	logical. if TRUE .txt files will be loaded as tables instead of .csv
`encoding`	character. files encoding.default is Latin-1
`stringsAsFactors`	logical: should character vectors be converted to factors? Note that this is overridden by as.is and colClasses, both of which allow finer control.
`header`	a logical value indicating whether the files contain the names of the variables as its first line. If missing, the value is determined from the file format: header is set to TRUE if and only if the first row contains one fewer field than the number of columns.
`quote`	the set of quoting characters. To disable quoting altogether, use quote = "". See scan for the behaviour on quotes embedded in quotes. Quoting is only considered for columns read as character, which is all of them unless colClasses is specified.
`fill`	logical. If TRUE then in case the rows have unequal length, blank fields are implicitly added.
`comment.char`	character: a character vector of length one containing a single character or an empty string. Use "" to turn off the interpretation of comments altogether.

Details

loadcsv_multi is used for uncompressed files in a single folder.it can be used either by entering the local directory the files are in, or just running it with no arguments for manual folder selection on windows OS. It receives some arguments from read.csv and they are listed in the arguments section. loadcsvfromZIP is used for comma separated tables inside of a .zip file. loadZIPcsvfromURL is used for comma separated tables inside of a .zip file on the internet, no download needed.

Value

A data.frame containing a representation of the data in the file.

See Also

loadcsvfromZIP loadcsv_multi

Index