Package 'saves'

February 20, 2015

Maintainer Gergely Daróczi <daroczig@rapporter.net>

License AGPL-3

BugReports https://github.com/daroczig/saves/issues

Title Fast load variables

LazyData no

Type Package

Author Gergely Daróczi <gergely@snowl.net>

Description The purpose of this package is to be able to save and load only the needed variables/columns of a dataframe in special binary files (tar archives) - which seems to be a lot faster method than loading the whole binary object (RData files) via load() function, or than loading columns from SQLite/MySQL databases via SQL commands (see vignettes). Performance gain on SSD drives is a lot more sensible compared to basic load() function. The performance improvement gained by loading only the chosen variables in binary format can be useful in some special cases (e.g. where merging data tables is not an option and very different datasets are needed for reporting), but be sure if using this package that you really need this, as non-standard file formats are used!

Version 0.5

URL http://r.snowl.net/saves/

Date 2013-12-27

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2013-12-27 16:07:34

R topics documented:

`evs.2000.hun` *European Values Survey (2000) - Hungary*

Description

A sample of European Values Survey (2000) dataset - Hungary

Format

A data frame with 1000 observations and 34 variables.

Details

This is only a sample dataset for package `ggpt` and `evs.2000.hun` does not contain all variables from the original dataset available to download from the EVS main site.

The data is made of quite a lot of variables, see below. The field work was taken in 1999 in Hungary. The cases represents the Hungarian population (n=1.000) and a weight variable is also given.

The included variables were choosen to be able to test the `ggpot` package's functions, and mainly includes 'multiple response' and some 'matrix/array type' questions.

Theese are answeres to the following questions:

- how important in your life: work (Q1A)
- how important in your life: family (Q1B)
- how important in your life: friends and acquaintances (Q1C)
- how important in your life: leisure time (Q1D)
- how important in your life: politics (Q1E)
- how important in your life: religion (Q1F)
- how often discuss politics with friends (Q2)
- taking all things together how happy are you (Q4)
- religious service important: birth (Q27A)
- religious service important: marriage (Q27B)
- religious service important: death (Q27C)
- are you a religious person (Q28)
- do you believe in: God (Q30A)
- do you believe in: life after death (Q30B)
- do you believe in: hell (Q30C)
- do you believe in: heaven (Q30D)
- do you believe in: sin (Q30E)
- do you believe in: telepathy (Q30F)
- do you believe in: reincarnation (Q31)

- how much confidence in: church (Q58A)
- how much confidence in: armed forces (Q58B)
- how much confidence in: education system (Q58C)
- how much confidence in: the press (Q58D)
- how much confidence in: trade unions (Q58E)
- how much confidence in: the police (Q58F)
- how much confidence in: parliament (Q58G)
- how much confidence in: civil service (Q58H)
- how much confidence in: social security system (Q58I)
- how much confidence in: european union (Q58J)
- how much confidence in: united nations organisation (Q58L)
- how much confidence in: health care system (Q58M)
- how much confidence in: justice system (Q58N)
- age completed education respondent (Q93)

References

http://www.europeanvaluesstudy.eu/

`loads` *Loading only given variables of a data.frame from binary file*

Description

`loads` does what the name suggests: it loads data from a special binary file format (RDatas) made up by the `saves` function. This special, uncompressed tar archive inlcudes several separate RData files (saved by `save` function) as being columns/variables of a data.frame.

Usage

```
loads(file = NULL, variables = NULL, to.data.frame = FALSE,
  ultra.fast = FALSE)
```

Arguments

`file`	character string: the (RDatas) filename from which to load the variables. If using `ultra.fast = TRUE` option, specify the directory holding the uncompressed R objects (saved via `saves(..., ultra.fast = TRUE)`).
`variables`	A character vector containing the variable names to load
`to.data.frame`	boolean: the default behavior of `loads` is to concatenate the variables to a list. This could be overriden with TRUE argument specified at to.data.frame parameter, which will return a dataframe instead of list. Only do this if all your variables have the same number of cases!

`ultra.fast` boolean: if TRUE, ultra fast (...) processing is done without any check to parameters or file existence/permissions. Be sure if using this setting as no debugging is done! Only recommended for servers dealing with lot of R objects' saves and loads in a monitored environment. Also, for performance gain, it is advised not to convert the list to data frame (to.data.frame = FALSE).

Details

The purpose of this function is to be able only a few variables of a data.frame really fast. It is done by reading and writing datas in binary format without any transformations, and combining the speed of only reading the needed part of an archive.

Some minor experiments shows a huge performance gain against using SQLite/MySQL backends or loading whole binary data, but be conscious always choosing the aprropriate method to write and read data.

The author of this package would like to emphasize: this package could be useful only in few cases!

Value

Loaded data.frame

See Also

`saves` to save R objects to RDatas binary format

Examples

```
## Not run:
# Loading the 'v1' and 'v5' variables of the demo dataset.
data(evs.2000.hun)
saves(evs.2000.hun)
evs.filtered.list <- loads("evs.2000.hun.RDatas", c('v1', 'v5'))
evs.filtered.df <- loads("evs.2000.hun.RDatas", c('v1', 'v5'), to.data.frame=TRUE)

## End(Not run)
```

---

saves *Save the variables of a data.frame in distinct binary files*

Description

`saves` does what the name suggests: it saves dataframe(s) or list(s) to disk in a special, binary format. This binary format consists of distinct binary files of all separate variables of a dataframe/list merged into an uncompressed tar archive. This is done via a loop, which saves each variable/column to an an external representation of the R objects via `save` in a temporary directory. Theese 'RData' files are archived to an 'RDatas' tar archive, uncompressed for better speed.

Usage

```
saves(..., list = character(), file = NULL, overwrite = FALSE,
  ultra.fast = FALSE)
```

Arguments

...	R objects: the names of the objects to be saved (as symbols or character strings)
list	character vector: the name(s) of the data frame(s) or list(s) to save
file	character vector: the (RDatas) filename(s) in which to save the variables in the current working directory
overwrite	boolean: if TRUE, existing files will be deleted before saving. Default set to FALSE, which will report error on conflicting file names.
ultra.fast	boolean: if TRUE, ultra fast (...) processing is done without any check to parameters, also no archiving or compression is done. Be sure if using this setting, as many uncompressed files could be generated in the working directory's subdirectory named to df. Only recommended for servers dealing with lot of R objects' saves and loads in a monitored environment.

Value

The saved filename(s) (invisible).

See Also

loads to load R objects from RDatas binary format

Examples

```
## Not run:
## Saving the demo dataset to evs.2000.hun.RDatas in current working directory.
data(evs.2000.hun)
saves(evs.2000.hun)
## Saving both the demo dataset and mtcars to current working directory
saves(evs.2000.hun, mtcars)
saves(list=c('evs.2000.hun', 'mtcars'))
## Saving all kind of cars :)
saves(cars, mtcars, overwrite = T)
saves(list=c('cars', 'mtcars'), overwrite = T)

## End(Not run)
```

Index