Package 'REQS'

May 7, 2026

Type Package

Title R/EQS Interface

Version 0.8-13

Date 2022-09-25

Description Contains the function run.eqs() which calls an EQS script file, executes the EQS estimation, and, finally, imports the results as R objects. These two steps can be performed separately: call.eqs() calls and executes EQS, whereas read.eqs() imports existing EQS outputs as objects into R. It requires EQS 6.2 (build 98 or higher).

Imports gtools

Depends R (>= 3.5.0)

License GPL-3

LazyLoad yes

NeedsCompilation no

Author Patrick Mair [aut, cre],
Eric Wu [aut]

Maintainer Patrick Mair <mair@fas.harvard.edu>

Repository CRAN

Date/Publication 2022-09-28 09:34:02 UTC

Contents

`call.eqs` *Call EQS from R*

Description

This function calls an EQS script file (.eqs) and executes it.

Usage

```
call.eqs(EQSpgm, EQSmodel, serial, Rmatrix = NA, datname = NA, LEN = 2000000)
```

Arguments

`EQSpgm`	String containing path including program name where EQS is located (see details)
`EQSmodel`	String containing path where .eqs script file is located (see details)
`serial`	EQS serial number as character
`Rmatrix`	Optional matrix argument if data or covariances are stored in R
`datname`	If `data` is specified, a filename (string) must be provided for saving the data in text format (blank separated; see details)
`LEN`	Integer containing number of working array units. By default, it is 2000000 8 bytes units

Details

If the path in `EQSpgm` and `EQSmodel` contains a blank, single quotes and double quotes are required in argument. See `EQSpgm` argument in examples. The last statement in the `EQSpgm` argument refers to the name of the executable program file. Under Windows it is `"...WINEQS"` (referring to WINEQS.exe), under Mac `"...MACEQS"` and under Linux `"...EQS"`. When specifying the path, use slash instead of backslash.

The .ETS, .CBK and .ETP files are written in the directory where the .eqs file is located. Unless another path is provided within in the .eqs script file.

The argument `datname` must match with the input data specified in the corresponding .eqs file.

Value

Returns `TRUE` is the estimation was succesfully and `FALSE` otherwise.

Author(s)

Patrick Mair, Eric Wu

References

Bentler, P. M. (1995). EQS Program Manual. Encino, CA: Multivariate Software Inc.

See Also

read.eqs, run.eqs

Examples

```
## Not run:
##not executable, valid serial number has to be provided
res <- call.eqs(EQSpgm = "C:/Program Files/EQS61/WINEQS.EXE",
                EQSmodel = "c:/eqs61/examples/manul7.eqs", serial = "1234")

## End(Not run)
```

---

read.eqs *Import of EQS outputs into R*

Description

This function reads EQS output files (.ets, .CBK and .ETP) into R and stores the results as objects.

Usage

```
read.eqs(file)
```

Arguments

file	The name (string) of the .ets file or the full path which the data are to be read from. If it does not contain an absolute path, the file name is relative to the current working directory, 'getwd()'. A .CBK and .ETP file have to be of the same name and in the same directory.

Details

The value list below provides objects for the full EQS output. If in EQS some objects are not computed, the corresponding values in R are NA.

Value

Returns a list with the following objects:

model.info	General model information
pval	p-values for various test statistics
fit.indices	Variuos fit indices
model.desc	Descriptive measures
Phi	Phi matrix
Gamma	Gamma matrix

`Beta`	Beta matrix
`par.table`	Parameter table (with standard errors)
`sample.cov`	Sample covariance matrix
`sigma.hat`	Model covariance matrix
`inv.infmat`	Inverse information matrix
`rinv.infmat`	Robust inverse information matrix
`cinv.infmat`	Corrected inverse information matrix
`derivatives`	First derivatives
`moment4`	Matrix with 4th moments
`ssolution`	Standardized elements
`Rsquared`	R-squared measures
`fac.means`	Factor means
`var.desc`	Descriptive measures for the variables (univariate statistics)
`indstd`	Independent variable standardization vector
`depstd`	Dependent variable standardization vector

Author(s)

Patrick Mair, Eric Wu

References

Bentler, P. M. (2008). EQS Program Manual. Encino, CA: Multivariate Software Inc.

See Also

`call.eqs`, `run.eqs`

Examples

```
## Not run:
##not executable
eqsout <- read.eqs("c:/home/user/eqs/eqsfile.ets")

## End(Not run)
```

run.eqs *Run EQS from R*

Description

Calls an EQS script file from R, executes EQS, and imports the results into R. Basically it is a wrapper function of `call.eqs` and the subsequent `read.eqs`.

Usage

```
run.eqs(EQSpgm, EQSmodel, serial, Rmatrix = NA, datname = NA, LEN = 2000000)
```

Arguments

`EQSpgm`	String containing path where EQS is located (see details)
`EQSmodel`	String containing path where .eqs script file is located (see details)
`serial`	EQS serial number as integer value
`Rmatrix`	Optional matrix argument if data or covariances are stored in R
`datname`	If `data` is specified, a filename (string) must be provided for saving the data in text format (blank separated; see details)
`LEN`	Integer containing number of working array units. By default, it is 2000000 8 bytes units

Details

If the path in `EQSpgm` and `EQSmodel` contains a blank, single quotes and double quotes are required in argument. See `EQSpgm` argument in examples. The last statement in the `EQSpgm` argument refers to the name of the executable program file. Under Windows it is `"...WINEQS"` (referring to WINEQS.exe), under Mac `"...MACEQS"` and under Linux `"...EQS"`. When specifying the path, use slash instead of backslash.

The .ETS, .CBK and .ETP files are written in the directory where the .eqs file is located. Note that these 3 files must be in the same directory than the .eqs file.

The argument `datname` must match with the input data specified in the corresponding .eqs file. This option can be used for simulations: Generate data in R, `run.eqs()` on with the corresponding `data` argument, pick out the relevant return values.

The value list below provides objects for the full EQS output. If in EQS some objects are not computed, the corresponding values in R are `NA`.

Value

Returns a list with the following objects:

`success`	`TRUE` if estimation was successful, `FALSE` otherwise
`model.info`	General model information

pval	p-values for various test statistics
fit.indices	Variuos fit indices
model.desc	Descriptive measures
Phi	Phi matrix
Gamma	Gamma matrix
Beta	Beta matrix
par.table	Parameter table (with standard errors)
sample.cov	Sample covariance matrix
sigma.hat	Model covariance matrix
inv.infmat	Inverse information matrix
rinv.infmat	Robust inverse information matrix
cinv.infmat	Corrected inverse information matrix
derivatives	First derivatives
moment4	Matrix with 4th moments
ssolution	Standardized elements
Rsquared	R-squared measures
fac.means	Factor means
var.desc	Descriptive measures for the variables (univariate statistics)
indstd	Independent variable standardization vector
depstd	Dependent variable standardization vector

Author(s)

Patrick Mair, Eric Wu

References

Bentler, P. M. (1995). EQS Program Manual. Encino, CA: Multivariate Software Inc.

See Also

read.eqs, call.eqs

Examples

```
## Not run:
##not executable, valid serial number has to be provided
res <- run.eqs(EQSpgm = "C:/Program Files/EQS61/WINEQS.EXE",
               EQSmodel = "c:/eqs61/examples/manul7.eqs", serial = "1234")

##For instance, to extract the parameter table you can do
res$par.table

##simulation example: not executable, provide serial number and proper eqs script file
```

```
##simulated 100 replications, extract CFI
cfivec <- NULL
for (i in 1:100) {
  X <- matrix(rnorm(1000), ncol = 10, nrow = 100)
  res <- run.eqs(EQSpgm = "C:/Program Files/EQS61/WINEQS.EXE",
                 EQSmodel = "c:/eqs61/examples/manul7.eqs", data = X,
                 datname = "manul7.dat", serial = "1234")
cfivec <- c(cfivec, res.run$fit.indices[9,])
}

## End(Not run)
```

Index