Package 'sasr'

March 31, 2025

Type Package

Title 'SAS' Interface

Version 0.1.5

Date 2025-03-30

Description Provides a 'SAS' interface, through 'SASPy'(<https://sassoftware.github.io/saspy/>) and 'reticulate'(<https://rstudio.github.io/reticulate/>). This package helps you create 'SAS' sessions, execute 'SAS' code in remote 'SAS' servers, retrieve execution results and log, and exchange datasets between 'SAS' and 'R'. It also helps you to install 'SASPy' and create a configuration file for the connection. Please review the 'SASPy' license file as instructed so that you comply with its separate and independent license.

License Apache License 2.0

URL https://github.com/insightsengineering/sasr/, https://insightsengineering.github.io/sasr/latest-tag/

BugReports https://github.com/insightsengineering/sasr/issues

Depends R (>= 3.6)

Imports checkmate, lifecycle, reticulate

Suggests knitr, mockery, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.2

NeedsCompilation no

Author Liming Li [aut, cre],
Daniel Sabanes Bove [aut],
Isaac Gravestock [aut],
F. Hoffmann-La Roche AG [cph, fnd],
AstraZeneca plc [cph, fnd]

Maintainer Liming Li <liming.li1@astrazeneca.com>
Repository CRAN
Date/Publication 2025-03-31 18:50:02 UTC

Contents

`sasr-package` `sasr` *Package*

Description

`sasr` provides interface to SAS through `saspy` and `reticulate` in R.

Author(s)

Maintainer: Liming Li <liming.li1@astrazeneca.com>

Authors:

- Daniel Sabanes Bove <daniel.sabanesbove@gmail.com>
- Isaac Gravestock <isaac.gravestock@roche.com>

Other contributors:

- F. Hoffmann-La Roche AG [copyright holder, funder]
- AstraZeneca plc [copyright holder, funder]

See Also

Useful links:

- https://github.com/insightsengineering/sasr/
- https://insightsengineering.github.io/sasr/latest-tag/
- Report bugs at https://github.com/insightsengineering/sasr/issues

df2sd *Transfer data.frame to SAS*

Description

[Experimental] Transfer `data.frame` object from R environment to SAS.

Usage

```
df2sd(df, table = "_df", libref = "", ..., sas_session = get_sas_session())
```

Arguments

`df`	(`data.frame`) data frame to be transferred.
`table`	(`character`) table name in SAS.
`libref`	(`character`) library name in SAS.
`...`	additional arguments for `saspy.sasbase.SASsession.df2sd`
`sas_session`	(`saspy.sasbase.SASsession`) SAS session.

Value

"saspy.sasdata.SASdata" object.

get_sas_cfg *Obtain the SAS Configuration File*

Description

[Experimental] Obtain the file path of the SAS configuration file.

Usage

```
get_sas_cfg()
```

Details

Obtain the default sas configuration file. By default, it will search the `sascfg_personal.py` file under current directory. If it does not exist, it will search this file under home directory. If this file does not exist, NULL will be returned.

Value

The file path of default SAS configuration file, or NULL if not found.

get_sas_session *Get the Last or Default SAS Session*

Description

[Experimental] Obtain the last session or default session.

Usage

```
get_sas_session()
```

Details

this function is designed to facilitate the R users programming practice of function oriented programming instead of object oriented programmings.

Value

A new SAS session if there are no previous SAS session, or the last SAS session created.

install_saspy *Install* saspy *Module*

Description

[Experimental] Install saspy module in reticulate.

Usage

```
install_saspy(method = "auto", conda = "auto")
```

Arguments

method	(character) method to install saspy.
conda	(character) path to conda executable.

Value

No return value.

run_sas *Run SAS code with SAS Session*

Description

[Experimental] Run SAS code with a SAS session.

Usage

```
run_sas(sas_code, results = c("TEXT", "HTML"), sas_session = get_sas_session())
```

Arguments

sas_code	(character) sas code to be executed.
results	(character) sas code execution results type.
sas_session	(saspy.sasbase.SASsession) SAS session.

Details

run_sas will run sas code through SAS session. The results is a named list of LST and LOG. The result part will be stored in LST, and log will be stored in LOG. If results argument is "TEXT", then results are in text format; if results argument is "HTML", then results are in html format.

Value

Named list with following elements:

- LOG: string of SAS execution log.
- LST: string of SAS execution result, in html or txt format.

sascfg *Create SAS Session Configuration File*

Description

[Experimental] Create SAS session configuration file based on argument.

Usage

```
sascfg(
  name = "default",
  host,
  saspath,
  ssh = system("which ssh", intern = TRUE),
  encoding = "latin1",
  options = list("-fullstimer"),
  ...,
  sascfg = "sascfg_personal.py"
)
```

Arguments

`name`	(`character`) name of the configuration.
`host`	(`character`) host name of remote server.
`saspath`	(`character`) SAS executable path on remote server.
`ssh`	(`character`) executable path of ssh.
`encoding`	(`character`) encoding of the SAS session.
`options`	(`list`) additional list of arguments to pass to `ssh` command.
`...`	additional arguments.
`sascfg`	(`character`) target file of configuration.

Details

`host` and `saspath` are required to connect to remote SAS server. Other arguments can follow default. If transferring datasets is needed and the client(running sasr) is not reachable from the server, then tunnelling is required. Use `tunnel = `, `rtunnel = ` to specify tunnels and reverse tunnels. The values should be length 1 integer.

Value

No return value.

sas_engine *SAS engine function*

Description

SAS engine function

Usage

```
sas_engine(options)
```

Arguments

options	See knitr documentation on engines.

sas_session *Create SAS Session Based on Configuration File*

Description

[Experimental] Create a SAS session.

Usage

```
sas_session(sascfg = get_sas_cfg(), ...)
```

Arguments

sascfg	(string) SAS session configuration.
...	additional arguments passed to saspy.SASsession(). Can override the configuration file.

Value

SAS session.

sas_session_ssh *Create SAS Session Based on Configuration File*

Description

[Deprecated]

Usage

```
sas_session_ssh(sascfg = get_sas_cfg(), ...)
```

Arguments

sascfg	(string) SAS session configuration.
...	additional arguments passed to saspy.SASsession(). Can override the configuration file.

Value

SAS session.

sd2df *Transfer SAS Data to R*

Description

[Experimental] Transfer the table in SAS session to R.

Usage

```
sd2df(table, libref = "", ..., sas_session = get_sas_session())
```

Arguments

table	(character) table name in SAS.
libref	(character) library name in SAS.
...	additional arguments for saspy.sasbase.SASsession.sd2df
sas_session	(saspy.sasbase.SASsession) SAS session.

Value

data.frame object.

Index