Package 'pysd2r'

May 9, 2026

Title API to 'Python' Library 'pysd'

Version 0.1.0

Description Using the R package 'reticulate', this package creates an interface to the 'pysd' toolset. The package provides an R interface to a number of 'pysd' functions, and can read files in 'Vensim' 'mdl' format, and 'xmile' format. The resulting simulations are returned as a 'tibble', and from
that the results can be processed using 'dplyr' and 'ggplot2'. The package has been tested using 'python3'.

License MIT + file LICENSE

Depends R (>= 3.3)

Encoding UTF-8

LazyData true

Imports knitr, reticulate, tibble

Suggests dplyr, ggplot2, testthat

RoxygenNote 6.1.0

VignetteBuilder knitr

SystemRequirements 'python3' needs to built for the same architecture
R is built for (32 or 64 bit).

NeedsCompilation no

Author Jim Duggan [aut, cre]

Maintainer Jim Duggan <jim.duggan@nuigalway.ie>

Repository CRAN

Date/Publication 2018-09-03 12:30:10 UTC

Contents

`get_doc` *Formats a table of variable names*

Description

`get_doc()` Get mode variable names

Usage

```
get_doc(o)
```

Arguments

`o`	is the ipysd S3 object

Value

tibble

Examples

```
## Not run:
target <- system.file("models/vensim", "Population.mdl", package = "pysd2r")
py <- pysd_connect()
py <- read_vensim(py, target)
mdoc <- get_doc(py)

## End(Not run)
```

get_final_time *Gets the final time from the model*

Description

get_timestep uses pysd to fetch the time step from the model

Usage

```
get_final_time(o)
```

Arguments

o	is the ipysd S3 object

Details

As it's a generic function, this call is dispatched to set_component.isdpy

Value

The finaltime

Examples

```
## Not run:
target <- system.file("models/vensim", "Population.mdl", package = "pysd2r")
py <- pysd_connect()
py <- read_vensim(py, target)
final_time <- get_final_time(py)

## End(Not run)
```

get_initial_time *Gets the initial time from the model*

Description

get_initial_time uses pysd to fetch the time step from the model

Usage

```
get_initial_time(o)
```

Arguments

o	is the ipysd S3 object

Details

As it's a generic function, this call is dispatched to set_component.isdpy

Value

The initial time

Examples

```
## Not run:
target <- system.file("models/vensim", "Population.mdl", package = "pysd2r")
py <- pysd_connect()
py <- read_vensim(py, target)
initial_time <- get_initial_time(py)

## End(Not run)
```

`get_python_info` *Gets the current python configuration for reticulate*

Description

`get_python_info` returns information on what version of python is being used with reticulate

Usage

```
get_python_info()
```

Value

python information

Examples

```
## Not run:
get_python_info()

## End(Not run)
```

get_timestep *Gets the time step (DT) from the model*

Description

get_timestep uses pysd to fetch the time step from the model

Usage

```
get_timestep(o)
```

Arguments

o	is the ipysd S3 object

Details

As it's a generic function, this call is dispatched to set_component.isdpy

Value

The simulation time step

Examples

```
## Not run:
target <- system.file("models/vensim", "Population.mdl", package = "pysd2r")
py <- pysd_connect()
py <- read_vensim(py, target)
time_step  <- get_timestep(py)

## End(Not run)
```

pysd_connect *Creates an object to facilitate interaction with pysd*

Description

pysd_connect returns a ipysd object to the calling program. This object will contain a link variable to pysd and will subsequently store a reference to the simulation model in pysd.

Usage

```
pysd_connect()
```

Details

Link to pysd

The result is used as a parameter for read_vensim() & read_xmile() functions

Value

An S3 object of class ipysd

Examples

```
## Not run:
py pysd_connect()

## End(Not run)
```

`read_vensim` *Loads a Vensim simulation file (mdl)*

Description

`read_vensim()` calls `pysd.read_vensim()` and stores the object for further use. This is a key object, as it relates to a model and it can support a number of functions (e.g. model run, parameter changes)

Usage

```
read_vensim(o, file)
```

Arguments

`o`	is the ipysd S3 object
`file`	is the filename and path for the Vensim mdl file that needs to be simulated

Details

The result is used as a parameter for simulation calls.

As it's a generic function, this call is dispatched to read_vensim.isdpy

Value

An S3 object of class ipysd that will contain a reference to the model

Examples

```
## Not run:
target <- system.file("models/vensim", "Population.mdl", package = "pysd2r")
py <- pysd_connect()
read_vensim(py, target)

## End(Not run)
```

`read_xmile` *Loads a XMILE simulation file (.xmile)*

Description

`read_xmile()` calls `pysd.read_xmile()` and stores the object for further use. This is a key object, as it relates to a model and it can support a number of functions (e.g. model run, parameter changes)

Usage

```
read_xmile(o, file)
```

Arguments

`o`	is the ipysd S3 object
`file`	is the filename and path for the Vensim mdl file that needs to be simulated

Details

The result is used as a parameter for simulation calls.

As it's a generic function, this call is dispatched to read_xmile.isdpy

Value

An S3 object of class ipysd that will contain a reference to the model

Examples

```
## Not run:
target <- system.file("models/xmile", "Population.xmile", package = "pysd2r")
py <- pysd_connect()
read_xmile(py, target)

## End(Not run)
```

reload_model *Reloads the model from original mdl file*

Description

`reload_model()` Reloads the model

Usage

```
reload_model(o)
```

Arguments

o	is the ipysd S3 object

Value

ipysd object

Examples

```
## Not run:
target <- system.file("models/vensim", "Population.mdl", package = "pysd2r")
py <- pysd_connect()
py <- read_vensim(py, target)
set_time_values(py,0,10,0.5)
py<-reload_model(py)

## End(Not run)
```

run_model *Runs a simulation model*

Description

`run_model()` calls `run` in pysd and returns all the simulation output in tidy data format (tibble)

Usage

```
run_model(o)
```

Arguments

o	is the ipysd S3 object

Details

As it's a generic function, this call is dispatched to run_model.isdpy

Value

tibble containing the simulation results

Examples

```
## Not run:
target <- system.file("models/vensim", "Population.mdl", package = "pysd2r")
py <- pysd_connect()
py <- read_vensim(py, target)
results <- run_model(py)

## End(Not run)
```

`set_components` *Changes a model parameter*

Description

`set_components()` calls `.set_components()` and changes a resulting parameter in the model

Usage

```
set_components(o, vals)
```

Arguments

`o`	is the ipysd S3 object
`vals`	contains a list with the parameter and value to be changed

Details

As it's a generic function, this call is dispatched to set_component.isdpy

Examples

```
## Not run:
target <- system.file("models/vensim", "Population.mdl", package = "pysd2r")
py <- pysd_connect()
py <- read_vensim(py, target)
results <- run_model(py)
l <- list("Growth Fraction"=0.02)
set_components(py,l)
out2 <- run_model(py)

## End(Not run)
```

`set_time_values` *Sets the initial time, final time, and timestep*

Description

`set_time_valuesl()` sets the simulation times and DT

Usage

```
set_time_values(o, init, final, DT)
```

Arguments

`o`	is the ipysd S3 object
`init`	is the initial time
`final`	is the final time
`DT`	is the time step

Examples

```
## Not run:
target <- system.file("models/vensim", "Population.mdl", package = "pysd2r")
py <- pysd_connect()
py <- read_vensim(py, target)
set_time_values(py,0,10,0.5)

## End(Not run)
```

Index