

Package ‘pharmr’

October 15, 2021

Encoding UTF-8

Version 0.33.1

Date 2021-10-13

Title Interface to the 'PharmPy' 'Pharmacometrics' Library

Maintainer Rikard Nordgren <rikard.nordgren@farmaci.uu.se>

Depends R (>= 3.3.0), altair (>= 4.0.0)

SystemRequirements Python (>= 3.6.0)

Imports reticulate (>= 1.19), utils

Suggests testthat, magrittr, here

NeedsCompilation no

Description Interface to the 'PharmPy' 'pharmacometrics' library. The 'Reticulate' package is used to interface Python from R.

Config/reticulate list(packages = list(list(package = ``altair"),
list(package = ``pharmPy-core")))

URL <https://github.com/pharmPy/pharmr>

BugReports <https://github.com/pharmPy/pharmr/issues>

License BSD_2_clause + file LICENSE

RoxygenNote 7.1.2

Author Rikard Nordgren [aut, cre, cph],
Stella Belin [aut, cph],
Mats O. Karlsson [sad],
Andrew C. Hooker [sad],
Sebastian Ueckert [sad] (<<https://orcid.org/0000-0002-3712-0255>>),
Simon Carter [rev],
Simon Buatois [rev],
João A. Abrantes [rev],
F. Hoffmann-La Roche Ltd. [fnd]

Repository CRAN

Date/Publication 2021-10-15 07:30:02 UTC

R topics documented:

add_covariate_effect	4
add_estimation_step	6
add_iiv	7
add_individual_parameter	9
add_iov	9
add_peripheral_compartment	10
append_estimation_step_options	11
calculate_eta_shrinkage	12
calculate_individual_parameter_statistics	13
calculate_individual_shrinkage	14
calculate_pk_parameters_statistics	14
check_pharmpy	15
convert_model	16
copy_model	16
create_joint_distribution	17
create_results	18
create_rng	19
evaluate_expression	19
fit	20
fix_parameters	21
fix_parameters_to	22
get_model_covariates	23
get_number_of_individuals	23
get_number_of_observations	24
get_number_of_observations_per_individual	25
get_observations	26
has_additive_error_model	27
has_combined_error_model	27
has_proportional_error_model	28
has_zero_order_absorption	29
install_pharmpy	30
load_example_model	30
plot_individual_predictions	31
plot_iofv_vs_iofv	31
predict_influential_individuals	32
predict_influential_outliers	32
predict_outliers	33
print_model_symbols	34
read_model	34
read_model_from_database	35
read_model_from_string	36
read_results	37
remove_error_model	37
remove_estimation_step	38
remove_iiv	39
remove_iov	40

remove_lag_time	40
remove_peripheral_compartment	41
reset_index	42
run_tool	42
sample_individual_estimates	43
sample_parameters_from_covariance_matrix	44
sample_parameters_uniformly	45
set_additive_error_model	46
set_bolus_absorption	47
set_combined_error_model	48
set_dtbs_error_model	49
set_estimation_step	50
set_first_order_absorption	51
set_first_order_elimination	51
set_iiv_on_ruv	52
set_initial_estimates	53
set_lag_time	54
set_michaelis_menten_elimination	54
set_mixed_mm_fo_elimination	55
set_name	56
set_ode_solver	57
set_peripheral_compartments	57
set_power_on_ruv	58
set_proportional_error_model	59
set_seq_zo_fo_absorption	60
set_transit_compartments	61
set_weighted_error_model	61
set_zero_order_absorption	62
set_zero_order_elimination	63
split_joint_distribution	64
summarize_modelfit_results	64
transform_etas_boxcox	65
transform_etas_john_draper	66
transform_etas_tdist	67
unfix_parameters	68
unfix_parameters_to	69
update_inits	69
update_source	70
use_thetas_for_error_stdev	71
write_model	72

add_covariate_effect *add_covariate_effect*

Description

Adds covariate effect to `:class:pharmpy.model`.

The following effects have templates:

- Linear function for continuous covariates (*lin*)
- Function:

math::

coveff = 1 + theta * (cov - median)

- Init: 0.001
- Upper:
- If median of covariate equals minimum: $:math:100,000$
- Otherwise: $:math:\frac{1}{\{\{median\} - \{min\}\}}$
- Lower:
- If median of covariate equals maximum: $:math:-100,000$
- Otherwise: $:math:\frac{1}{\{\{median\} - \{max\}\}}$
- Linear function for categorical covariates (*cat*)
- Function:
- If covariate is most common category:

math::

coveff = 1

- For each additional category:

math::

coveff = 1 + theta

- Init: $:math:0.001$
- Upper: $:math:100,000$
- Lower: $:math:-100,000$
- Piecewise linear function/"hockey-stick", continuous covariates only (*piece_lin*)
- Function:
- If cov \leq median:

math::

coveff = 1 + theta1 * (cov - median)

- If cov > median:

math::

$$\text{coveff} = 1 + \text{theta2} * (\text{cov} - \text{median})$$

- Init: 0.001
- Upper:
- For first state: $\frac{1}{\{\text{median}\} - \{\text{min}\}}$
- Otherwise: $100,000$
- Lower:
- For first state: $-100,000$
- Otherwise: $\frac{1}{\{\text{median}\} - \{\text{max}\}}$
- Exponential function, continuous covariates only (*exp*)
- Function:

math::

$$\text{coveff} = \exp(\text{theta} * (\text{cov} - \text{median}))$$

- Init:
- If lower > 0.001 or upper < 0.001: $\frac{\{\text{upper}\} - \{\text{lower}\}}{2}$
- If estimated init is 0: $\frac{\{\text{upper}\}}{2}$
- Otherwise: 0.001
- Upper:
- If min - median = 0 or max - median = 0: 100
- Otherwise:

math::

$$\min(\text{fraclog}(0.01)\text{min} - \text{median}, \text{fraclog}(100)\text{max} - \text{median})$$

- Lower:
- If min - median = 0 or max - median = 0: 0.01
- Otherwise:

math::

$$\max(\text{fraclog}(0.01)\text{max} - \text{median}, \text{fraclog}(100)\text{min} - \text{median})$$

- Power function, continuous covariates only (*pow*)
- Function:

math::

$$\text{coveff} = (\text{fraccovmedian})^{\text{theta}}$$

- Init: 0.001
- Upper: $100,000$
- Lower: -100

Usage

```
add_covariate_effect(model, parameter, covariate, effect, operation = "*")
```

Arguments

model	(Model) PharmPy model to add covariate effect to.
parameter	(str) Name of parameter to add covariate effect to.
covariate	(str) Name of covariate.
effect	(str) Type of covariate effect. May be abbreviated covariate effect (see above) or custom.
operation	(str, optional) Whether the covariate effect should be added or multiplied (default).

Value

(Model) Reference to the same model

Examples

```
## Not run:
model <- load_example_model("pheno")
add_covariate_effect(model, "CL", "APGR", "exp")
model$statements$full_expression_from_odes("CL")

## End(Not run)
```

```
add_estimation_step    add_estimation_step
```

Description

Add estimation step

Adds estimation step for a model in a given index. Methods currently supported are: FO, FOCE, ITS, LAPLACE, IMPMAP, IMP, SAEM

Usage

```
add_estimation_step(
  model,
  method,
  interaction = TRUE,
  options = NULL,
  idx = NULL
)
```

Arguments

model	(Model) Pharmpy model
method	(str) estimation method to change to
interaction	(logical) whether to use interaction or not, default is true
options	(list) any additional tool specific options
idx	(integer) index of estimation step, default is NULL (adds step at the end)

Value

(Model) Reference to the same model object

See Also

set_estimation_step
 remove_estimation_step
 append_estimation_step_options

Examples

```
## Not run:
model <- load_example_model("pheno")
opts <- list('NITER'=1000, 'ISAMPLE'=100, 'EONLY'=1)
add_estimation_step(model, "IMP", options=opts)
ests <- model$estimation_steps
len(ests)
ests[2]

## End(Not run)
```

 add_iiv

 add_iiv

Description

Adds IIVs to :class:pharmpy.model.

Effects that currently have templates are:

- Additive (*add*)
- Proportional (*prop*)
- Exponential (*exp*)
- Logit (*logit*)

For all except exponential the operation input is not needed. Otherwise user specified input is supported. Initial estimates for new etas are 0.09.

Usage

```
add_iiv(  
  model,  
  list_of_parameters,  
  expression,  
  operation = "*",  
  eta_names = NULL  
)
```

Arguments

model	(Model) Pharmpy model to add new IIVs to.
list_of_parameters	(str, vector) Name/names of parameter to add new IIVs to.
expression	(str, vector) Effect/effects on eta. Either abbreviated (see above) or custom.
operation	(str, vector, optional) Whether the new IIV should be added or multiplied (default).
eta_names	(str, vector, optional) Custom name/names of new eta

Value

(Model) Reference to the same model

See Also

add_iov
remove_iiv
remove_iov

Examples

```
## Not run:  
model <- load_example_model("pheno")  
remove_iiv(model, "CL")  
add_iiv(model, "CL", "add")  
model$statements$find_assignment("CL")  
  
## End(Not run)
```

```
add_individual_parameter  
    add_individual_parameter
```

Description

Add an individual or pk parameter to a model

Usage

```
add_individual_parameter(model, name)
```

Arguments

model	(Model) Pharnpy model
name	(str) Name of individual/pk parameter

Value

(Model) Reference to same model

Examples

```
## Not run:  
model <- load_example_model("pheno")  
add_individual_parameter(model, "KA")  
model$statements$find_assignment("KA")  
  
## End(Not run)
```

```
add_iov    add_iov
```

Description

Adds IOVs to :class:pharnpy.model.

Initial estimate of new IOVs are 10% of the IIV eta it is based on.

Usage

```
add_iov(model, occ, list_of_parameters = NULL, eta_names = NULL)
```

Arguments

model	(Model) PharmPy model to add new IOVs to.
occ	(str) Name of occasion column.
list_of_parameters	(str, vector) List of names of parameters and random variables. Accepts random variable names, parameter names, or a mix of both.
eta_names	(str, vector) Custom names of new etas. Must be equal to the number of input etas times the number of categories for occasion.

Value

(Model) Reference to the same model

See Also

add_iiv
 remove_iiv
 remove_iov

Examples

```
## Not run:
model <- load_example_model("pheno")
add_iov(model, "TIME", "CL")
model$statements$find_assignment("CL")

## End(Not run)
```

```
add_peripheral_compartment
      add_peripheral_compartment
```

Description

Add a peripheral distribution compartment to model

The rate of flow from the central to the peripheral compartment will be parameterized as Q_{Pn} / VC where VC is the volume of the central compartment. The rate of flow from the peripheral to the central compartment will be parameterized as Q_{Pn} / VP_n where VP_n is the volume of the added peripheral compartment.

Initial estimates:

```
===== n =====
1 :math:{CL} = {CL'}, :math:{VC} = {VC'}, :math:{QP1} = {CL'} and :math:{VP1} = {VC'} * 0.05
2 :math:{QP1} = {QP1' / 2}, :math:{VP1} = {VP1'}, :math:{QP2} = {QP1' / 2} and :math:{VP2} = {VP1'}
```

Usage

```
add_peripheral_compartment(model)
```

Arguments

model (Model) PharmPy model

Value

(Model) Reference to same model

See Also

set_peripheral_compartment
remove_peripheral_compartment

Examples

```
## Not run:  
model <- load_example_model("pheno")  
add_peripheral_compartment(model)  
model$statements$ode_system  
  
## End(Not run)
```

```
append_estimation_step_options  
    append_estimation_step_options
```

Description

Append estimation step options
Appends options to an existing estimation step.

Usage

```
append_estimation_step_options(model, options, idx)
```

Arguments

model (Model) PharmPy model
options (list) any additional tool specific options
idx (integer) index of estimation step

Value

(Model) Reference to the same model object

See Also

add_estimation_step
set_estimation_step
remove_estimation_step

Examples

```
## Not run:  
model <- load_example_model("pheno")  
opts <- list('NITER'=1000, 'ISAMPLE'=100, 'EONLY'=1)  
append_estimation_step_options(model, options=opts, idx=0)  
est <- model$estimation_steps[1]  
len(est$options)  
  
## End(Not run)
```

calculate_eta_shrinkage
calculate_eta_shrinkage

Description

Calculate eta shrinkage for each eta

Usage

```
calculate_eta_shrinkage(model, sd = FALSE)
```

Arguments

model	(Model) PharmPy model
sd	(logical) Calculate shrinkage on the standard deviation scale (default is to calculate on the variance scale)

Value

(Series) Shrinkage for each eta

See Also

calculate_individual_shrinkage

Examples

```
## Not run:
model <- load_example_model("pheno")
calculate_eta_shrinkage(model)
calculate_eta_shrinkage(model, sd=TRUE)

## End(Not run)
```

```
calculate_individual_parameter_statistics
      calculate_individual_parameter_statistics
```

Description

Calculate statistics for individual parameters

Calculate the mean (expected value of the distribution), variance (variance of the distribution) and standard error for individual parameters described by arbitrary expressions. Any dataset column or variable used in the model can be used in the expression. The exception being that variables that depends on the solution of the ODE system cannot be used. If covariates are used in the expression the statistics of the parameter is calculated at the median value of each covariate as well as at the 5:th and 95:th percentiles. If no parameter uncertainty is available for the model the standard error will not be calculated.

Usage

```
calculate_individual_parameter_statistics(model, exprs, rng = NULL)
```

Arguments

model	(Model) A previously estimated model
exprs	(str, sympy expression or iterable of str or sympy expressions) Expressions or equations for parameters of interest. If equations are used the names of the left hand sides will be used as the names of the parameters.
rng	(Generator or integer) Random number generator or integer seed

Value

(data.frame) A DataFrame of statistics indexed on parameter and covariate value.

Examples

```
## Not run:
model <- load_example_model("pheno")
rng <- create_rng(23)
calculate_individual_parameter_statistics(model, "K=CL/V", rng=rng)

## End(Not run)
```

```
calculate_individual_shrinkage  
    calculate_individual_shrinkage
```

Description

Calculate the individual eta-shrinkage

Definition: $ieta_shr = (\text{var}(\eta) / \omega)$

Usage

```
calculate_individual_shrinkage(model)
```

Arguments

model (Model) PharmPy model

Value

(DataFrame) Shrinkage for each eta and individual

See Also

calculate_eta_shrinkage

Examples

```
## Not run:  
model <- load_example_model("pheno")  
calculate_individual_shrinkage(model)  
  
## End(Not run)
```

```
calculate_pk_parameters_statistics  
    calculate_pk_parameters_statistics
```

Description

Calculate statistics for common pharmacokinetic parameters

Calculate the mean (expected value of the distribution), variance (variance of the distribution) and standard error for some individual pre-defined pharmacokinetic parameters.

Usage

```
calculate_pk_parameters_statistics(model, rng = NULL)
```

Arguments

model (Model) A previously estimated model
rng (Generator or integer) Random number generator or seed

Value

(data.frame) A DataFrame of statistics indexed on parameter and covariate value.

See Also

calculate_individual_parameter_statistics : Calculation of statistics for arbitrary parameters

Examples

```
## Not run:  
model <- load_example_model("pheno")  
rng <- create_rng(23)  
calculate_pk_parameters_statistics(model, rng=rng)  
  
## End(Not run)
```

check_pharmpy *Checks Pharmr and Pharmpy version.*

Description

Compares Pharmr version to which version of Pharmpy that is used.

Usage

```
check_pharmpy(pharmpy_version)
```

Arguments

pharmpy_version
Pharmpy version (can be called by pharmpy\$`__version__`)

Details

The documentation of modeling functions in Pharmr is automatically generated from the latest release of Pharmpy. This function checks if the two have the same major and minor version, otherwise the documentation may be outdated and an update is recommended.

convert_model	<i>convert_model</i>
---------------	----------------------

Description

Convert model to other format

Usage

```
convert_model(model, to_format)
```

Arguments

model	(Model) Model to convert
to_format	(str) Name of format to convert into. Currently supported 'nlmixr' and 'nonmem'

Value

(Model) New model object with new underlying model format

Examples

```
## Not run:  
model <- load_example_model("pheno")  
converted_model <- convert_model(model, "nlmixr")  
  
## End(Not run)
```

copy_model	<i>copy_model</i>
------------	-------------------

Description

Copies model to a new model object

Usage

```
copy_model(model, name = NULL)
```

Arguments

model	(Model) PharmPy model
name	(str) Optional new name of model

Value

(Model) A copy of the input model

Examples

```
## Not run:  
model <- load_example_model("pheno")  
model_copy <- copy_model(model, "pheno2")  
  
## End(Not run)
```

```
create_joint_distribution  
      create_joint_distribution
```

Description

Combines some or all etas into a joint distribution.

The etas must be IIVs and cannot be fixed. Initial estimates for covariance between the etas is dependent on whether the model has results from a previous results. In that case, the correlation will be calculated from individual estimates, otherwise correlation will be set to 10%.

Usage

```
create_joint_distribution(model, rvs = NULL)
```

Arguments

model	(Model) Pharmpy model
rvs	(vector) Sequence of etas or names of etas to combine. If NULL, all etas that are IIVs and non-fixed will be used (full block). NULL is default.

Value

(Model) Reference to the same model

See Also

split_joint_distribution : split etas into separate distributions

Examples

```
## Not run:
model <- load_example_model("pheno")
model$random_variables$etas
create_joint_distribution(model, c('ETA(1)', 'ETA(2)'))
model$random_variables$etas

## End(Not run)
```

create_results	<i>create_results</i>
----------------	-----------------------

Description

Create/recalculate results object given path to run directory

Usage

```
create_results(path, ...)
```

Arguments

path	(str, Path) Path to run directory
...	Arguments to pass to tool specific create results function

Value

(Results object for tool)

See Also

read_results

Examples

```
## Not run:
res <- create_results("frem_dir1")

## End(Not run)
```

create_rng	<i>create_rng</i>
------------	-------------------

Description

Create a new random number generator

PharmPy functions that use random sampling take a random number generator or seed as input. This function can be used to create a default new random number generator.

Usage

```
create_rng(seed)
```

Arguments

seed	(integer or rng) Seed for the random number generator or NULL (default) for a randomized seed. If seed is generator it will be passed through.
------	--

Value

(Generator) Initialized numpy random number generator object

Examples

```
## Not run:  
rng <- create_rng(23)  
rng$standard_normal()  
  
## End(Not run)
```

evaluate_expression	<i>evaluate_expression</i>
---------------------	----------------------------

Description

Evaluate expression using model

Calculate the value of expression for each data record. The expression can contain dataset columns, variables in model and population parameters. If the model has parameter estimates these will be used. Initial estimates will be used for non-estimated parameters.

Usage

```
evaluate_expression(model, expression)
```

Arguments

model (Model) PharmPy model
 expression (str or sympy expression) Expression to evaluate

Value

(data.frame) A series of one evaluated value for each data record

Examples

```
## Not run:
model <- load_example_model("pheno")
evaluate_expression(model, "TVCL*1000")

## End(Not run)
```

 fit

fit

Description

Fit models.

Usage

```
fit(models)
```

Arguments

models (vector) List of models or one single model

Value

(Model) Reference to same model

See Also

run_tool

Examples

```
## Not run:
model <- load_example_model("pheno")
fit(model)

## End(Not run)
```

fix_parameters	<i>fix_parameters</i>
----------------	-----------------------

Description

Fix parameters

Fix all listed parameters

Usage

```
fix_parameters(model, parameter_names)
```

Arguments

model (Model) Pharnpy model

parameter_names
(vector or str) one parameter name or a vector of parameter names

Value

(Model) Reference to the same model object

See Also

fix_parameters_to : Fixing and setting parameter initial estimates in the same function

unfix_paramaters : Unfixing parameters

unfix_paramaters_to : Unfixing parameters and setting a new initial estimate in the same function

Examples

```
## Not run:  
model <- load_example_model("pheno")  
model$parameters['THETA(1)']  
fix_parameters(model, 'THETA(1)')  
model$parameters['THETA(1)']  
  
## End(Not run)
```

fix_parameters_to *fix_parameters_to*

Description

Fix parameters to

Fix all listed parameters to specified value/values

Usage

```
fix_parameters_to(model, parameter_names, values)
```

Arguments

model (Model) Pharnpy model

parameter_names

(vector or str) one parameter name or a vector of parameter names

values

(vector or numeric) one value or a vector of values (must be equal to number of parameter_names)

Value

(Model) Reference to the same model object

See Also

fix_parameters : Fix parameters

unfix_paramaters : Unfixing parameters

unfix_paramaters_to : Unfixing parameters and setting a new initial estimate in the same function

Examples

```
## Not run:
model <- load_example_model("pheno")
model$parameters['THETA(1)']
fix_parameters_to(model, 'THETA(1)', 0.5)
model$parameters['THETA(1)']

## End(Not run)
```

get_model_covariates *get_model_covariates*

Description

List of covariates used in model

A covariate in the model is here defined to be a data item affecting the model prediction excluding dosing items.

Usage

```
get_model_covariates(model, strings = FALSE)
```

Arguments

model	(Model) PharmPy model
strings	(logical) Return strings instead of symbols? FALSE (default) will give symbols

Value

(vector) Covariate symbols or names

Examples

```
## Not run:  
model <- load_example_model("pheno")  
get_model_covariates(model)  
get_model_covariates(model, strings=TRUE)  
  
## End(Not run)
```

get_number_of_individuals
get_number_of_individuals

Description

Retrieve the number of individuals in the model dataset

Usage

```
get_number_of_individuals(model)
```

Arguments

model	(Model) PharmPy model
-------	-----------------------

Value

(integer) Number of individuals in the model dataset

Note

For NONMEM models this is the number of individuals of the active dataset, i.e. after filtering of IGNORE and ACCEPT and removal of individuals with no observations.

See Also

`get_number_of_observations` : Get the number of observations in a dataset

`get_number_of_observations_per_individual` : Get the number of observations per individual in a dataset

Examples

```
## Not run:  
model <- load_example_model("pheno")  
get_number_of_individuals(model)  
  
## End(Not run)
```

```
get_number_of_observations  
      get_number_of_observations
```

Description

Retrieve the total number of observations in the model dataset

Usage

```
get_number_of_observations(model)
```

Arguments

`model` (Model) PharmPy model

Value

(integer) Number of observations in the model dataset

Note

For NONMEM models this is the number of observations of the active dataset, i.e. after filtering of IGNORE and ACCEPT and removal of individuals with no observations.

See Also

`get_number_of_individuals` : Get the number of individuals in a dataset
`get_number_of_observations_per_individual` : Get the number of observations per individual in a dataset

Examples

```
## Not run:  
model <- load_example_model("pheno")  
get_number_of_observations(model)  
  
## End(Not run)
```

`get_number_of_observations_per_individual`
get_number_of_observations_per_individual

Description

Number of observations for each individual

Usage

```
get_number_of_observations_per_individual(model)
```

Arguments

`model` (Model) PharmPy model

Value

(data.frame) Number of observations in the model dataset

Note

For NONMEM models this is the individuals and number of observations of the active dataset, i.e. after filtering of IGNORE and ACCEPT and removal of individuals with no observations.

See Also

`get_number_of_individuals` : Get the number of individuals in a dataset
`get_number_of_observations_per_individual` : Get the number of observations per individual in a dataset

Examples

```
## Not run:  
model <- load_example_model("pheno")  
get_number_of_observations_per_individual(model)  
  
## End(Not run)
```

get_observations	<i>get_observations</i>
------------------	-------------------------

Description

Get observations from dataset

Usage

```
get_observations(model)
```

Arguments

model (Model) PharmPy model

Value

(data.frame) Observations indexed over ID and TIME

See Also

get_number_of_observations
get_number_of_observations_per_individual

Examples

```
## Not run:  
model <- load_example_model("pheno")  
get_observations(model)  
  
## End(Not run)
```

`has_additive_error_model`
has_additive_error_model

Description

Check if a model has an additive error model

Usage

```
has_additive_error_model(model)
```

Arguments

`model` (Model) The model to check

Value

(logical) TRUE if the model has an additive error model and FALSE otherwise

See Also

`has_proportional_error_model` : Check if a model has a proportional error model

`has_combined_error_model` : Check if a model has a combined error model

Examples

```
## Not run:  
model <- load_example_model("pheno")  
has_additive_error_model(model)  
  
## End(Not run)
```

`has_combined_error_model`
has_combined_error_model

Description

Check if a model has a combined additive and proportional error model

Usage

```
has_combined_error_model(model)
```

Arguments

model (Model) The model to check

Value

(logical) TRUE if the model has a combined error model and FALSE otherwise

See Also

`has_additive_error_model` : Check if a model has an additive error model

`has_proportional_error_model` : Check if a model has a proportional error model

Examples

```
## Not run:  
model <- load_example_model("pheno")  
has_combined_error_model(model)  
  
## End(Not run)
```

`has_proportional_error_model`
has_proportional_error_model

Description

Check if a model has a proportional error model

Usage

```
has_proportional_error_model(model)
```

Arguments

model (Model) The model to check

Value

(logical) TRUE if the model has a proportional error model and FALSE otherwise

See Also

`has_additive_error_model` : Check if a model has an additive error model

`has_combined_error_model` : Check if a model has a combined error model

Examples

```
## Not run:  
model <- load_example_model("pheno")  
has_proportional_error_model(model)  
  
## End(Not run)
```

```
has_zero_order_absorption  
      has_zero_order_absorption
```

Description

Check if ode system describes a zero order absorption
currently defined as having Infusion dose with rate not in dataset

Usage

```
has_zero_order_absorption(model)
```

Arguments

model (Model) Pharnpy model

Value

(Model) Reference to same model

Examples

```
## Not run:  
model <- load_example_model("pheno")  
has_zero_order_absorption(model)  
  
## End(Not run)
```

install_pharmpy	<i>Install Pharmpy</i>
-----------------	------------------------

Description

Install the pharmpy-core python package into virtual environment.

Usage

```
install_pharmpy(envname = "r-reticulate", method = "auto")
```

Arguments

envname	(str) name of environment. Default is r-reticulate
method	(str) type of environment type (virtualenv, conda). Default is auto (virtualenv is not available on Windows)

load_example_model	<i>load_example_model</i>
--------------------	---------------------------

Description

Load an example model

Load an example model from models built into Pharmpy

Usage

```
load_example_model(name)
```

Arguments

name	(str) Name of the model. Currently available model is "pheno"
------	---

Value

(Model) Loaded model object

Examples

```
## Not run:  
model <- load_example_model("pheno")  
model$statements  
  
## End(Not run)
```

plot_individual_predictions
plot_individual_predictions

Description

Plot DV and predictions grouped on individuals

Usage

```
plot_individual_predictions(model, predictions = NULL, individuals = NULL)
```

Arguments

model (Model) Previously run PharmPy model.
predictions (vector) A vector of names of predictions to plot. NULL for all available
individuals (vector) A vector of individuals to include. NULL for all individuals

Value

(alt.Chart) Plot

plot_iofv_vs_iofv *plot_iofv_vs_iofv*

Description

Plot individual OFV of two models against each other

Usage

```
plot_iofv_vs_iofv(model, other)
```

Arguments

model (Model) The first model
other (Model) The second model

Value

(alt.Chart) Scatterplot

predict_influential_individuals
predict_influential_individuals

Description

Predict influential individuals for a model using a machine learning model.

Usage

```
predict_influential_individuals(model)
```

Arguments

model (Model) PharmPy model

Value

(pd.DataFrame) Dataframe over the individuals with a dofV column containing the raw predicted delta-OFV and an influential column with a boolean to tell whether the individual is influential or not.

See Also

predict_influential_outliers
predict_outliers

predict_influential_outliers
predict_influential_outliers

Description

Predict influential outliers for a model using a machine learning model.

Usage

```
predict_influential_outliers(model)
```

Arguments

model (Model) PharmPy model

Value

(pd.DataFrame) Dataframe over the individuals with a outliers and dofV columns containing the raw predictions and influential, outlier and influential_outlier boolean columns.

See Also

predict_influential_individuals
predict_outliers

predict_outliers *predict_outliers*

Description

Predict outliers for a model using a machine learning model.

See the :ref:simeval <Individual OFV summary> documentation for a definition of the residual

Usage

```
predict_outliers(model)
```

Arguments

model (Model) PharmPy model

Value

(pd.DataFrame) Dataframe over the individuals with a `residual` column containing the raw predicted residuals and a `outlier` column with a boolean to tell whether the individual is an outlier or not.

See Also

predict_influential_individuals
predict_influential_outliers

Examples

```
## Not run:  
model <- load_example_model("pheno")  
predict_outliers(model)  
  
## End(Not run)
```

print_model_symbols *print_model_symbols*

Description

Print all symbols defined in a model

Symbols will be in one of the categories thetas, etas, omegas, epsilons, sigmas, variables and data columns

Usage

```
print_model_symbols(model)
```

Arguments

model (Model) PharmPy model object

Examples

```
## Not run:  
model <- load_example_model("pheno")  
print_model_symbols(model)  
  
## End(Not run)
```

read_model *read_model*

Description

Read model from file

Usage

```
read_model(path)
```

Arguments

path (str or Path) Path to model

Value

(Model) Read model object

See Also

read_model_from_database : Read model from database
read_model_from_string : Read model from string

Examples

```
## Not run:  
model <- read_model("/home/run1$mod")  
  
## End(Not run)
```

read_model_from_database
read_model_from_database

Description

Read model from model database

Usage

```
read_model_from_database(name, database = NULL)
```

Arguments

name (str) Name of model to use as lookup
database (Database) Database to use. Will use default database if not specified.

Value

(Model) Read model object

See Also

read_model : Read model from file
read_model_from_string : Read model from string

Examples

```
## Not run:  
model <- read_model_from_database("run1")  
  
## End(Not run)
```

read_model_from_string
read_model_from_string

Description

Read model from the model code in a string

Usage

```
read_model_from_string(code)
```

Arguments

code (str) Model code to read

Value

(Model) Read model object

See Also

read_model : Read model from file
read_model_from_database : Read model from database

Examples

```
## Not run:  
s <- '$PROBLEM  
$INPUT ID DV TIME  
$DATA file$csv  
$PRED  
Y=THETA(1)+ETA(1)+ERR(1)  
$THETA 1  
$OMEGA 0.1  
$SIGMA 1  
$ESTIMATION METHOD=1''  
read_model_from_string(s)  
  
## End(Not run)
```

read_results	<i>read_results</i>
--------------	---------------------

Description

Read results object from file

Usage

```
read_results(path)
```

Arguments

path (str, Path) Path to results file

Value

(Results object for tool)

See Also

create_results

Examples

```
## Not run:  
res <- read_results("results$json")  
  
## End(Not run)
```

remove_error_model	<i>remove_error_model</i>
--------------------	---------------------------

Description

Remove error model.

Usage

```
remove_error_model(model)
```

Arguments

model (Model) Remove error model for this model

Value

(Model) Reference to the same model object

Examples

```
## Not run:  
model <- load_example_model("pheno")  
model$statements$find_assignment("Y")  
remove_error_model(model)  
model$statements$find_assignment("Y")  
  
## End(Not run)
```

```
remove_estimation_step  
    remove_estimation_step
```

Description

Remove estimation step

Usage

```
remove_estimation_step(model, idx)
```

Arguments

model	(Model) PharmPy model
idx	(integer) index of estimation step to remove

Value

(Model) Reference to the same model object

See Also

add_estimation_step
set_estimation_step
append_estimation_step_options

Examples

```
## Not run:
model <- load_example_model("pheno")
opts <- list('NITER'=1000, 'ISAMPLE'=100, 'EONLY'=1)
add_estimation_step(model, "IMP", options=opts)
remove_estimation_step(model, 1)
ests <- model$estimation_steps
len(ests)

## End(Not run)
```

remove_iiv

remove_iiv

Description

Removes all IIV etas given a vector with eta names and/or parameter names.

Usage

```
remove_iiv(model, to_remove = NULL)
```

Arguments

model	(Model) Pharnpy model to create block effect on.
to_remove	(str, vector) Name/names of etas and/or name/names of individual parameters to remove. If NULL, all etas that are IIVs will be removed. NULL is default.

Value

(Model) Reference to the same model

See Also

```
remove_iov
add_iiv
add_iov
```

Examples

```
## Not run:
model <- load_example_model("pheno")
remove_iiv(model)
model$statements$find_assignment("CL")
model <- load_example_model("pheno")
remove_iiv(model, "V")
model$statements$find_assignment("V")

## End(Not run)
```

remove_iov	<i>remove_iov</i>
------------	-------------------

Description

Removes all IOV etas

Usage

```
remove_iov(model)
```

Arguments

model (Model) PharmPy model to remove IOV from.

Value

(Model) Reference to the same model

See Also

add_iiv

add_iov

remove_iiv

Examples

```
## Not run:  
model <- load_example_model("pheno")  
remove_iov(model)  
  
## End(Not run)
```

remove_lag_time	<i>remove_lag_time</i>
-----------------	------------------------

Description

Remove lag time from the dose compartment of model.

Usage

```
remove_lag_time(model)
```

Arguments

model (Model) PharmPy model

Value

(Model) Reference to same model

See Also

set_transit_compartments

set_lag_time

Examples

```
## Not run:
model <- load_example_model("pheno")
remove_lag_time(model)

## End(Not run)
```

```
remove_peripheral_compartment
      remove_peripheral_compartment
```

Description

Remove a peripheral distribution compartment from model

Initial estimates:

```
=====  
2 :math:{CL} = {CL'}, :math:{QP1} = {CL}' \text{ and } :math:{VP1} = {VC}' * 0.05 \text{ 3 } :math:{QP1} = ({QP1}' + {QP2}') / 2,  
:math:{VP1} = {VP1}' + {VP2}' \text{ }=====  
n
```

Usage

```
remove_peripheral_compartment(model)
```

Arguments

model (Model) PharmPy model

Value

(Model) Reference to same model

See Also

set_peripheral_compartment

add_peripheral_compartment

Examples

```
## Not run:
model <- load_example_model("pheno")
set_peripheral_compartments(model, 2)
remove_peripheral_compartment(model)
model$statements$ode_system

## End(Not run)
```

reset_index

Reset index

Description

Reset index of dataframe.

Reset index from a multi indexed data.frame so that index is added as columns

Usage

```
reset_index(df)
```

Arguments

df A data.frame converted from python using reticulate

run_tool

run_tool

Description

Run tool workflow

Usage

```
run_tool(name, ...)
```

Arguments

name (str) Name of tool to run
... Arguments to pass to tool

Value

(Results object for tool)

Examples

```
## Not run:  
model <- load_example_model("pheno")  
res <- run_tool("resmod", model)  
  
## End(Not run)
```

```
sample_individual_estimates  
      sample_individual_estimates
```

Description

Sample individual estimates given their covariance.

Usage

```
sample_individual_estimates(  
  model,  
  parameters = NULL,  
  samples_per_id = 100,  
  rng = NULL  
)
```

Arguments

model	(Model) Pharnpy model
parameters	(vector) A vector of a subset of individual parameters to sample. Default is NULL, which means all.
samples_per_id	(integer) Number of samples per individual
rng	(rng or integer) Random number generator or seed

Value

(data.frame) Pool of samples in a DataFrame

See Also

sample_parameters_from_covariance_matrix : Sample parameter vectors using the uncertainty covariance matrix

sample_parameters_uniformly : Sample parameter vectors using uniform distribution

Examples

```
## Not run:
model <- load_example_model("pheno")
rng <- create_rng(23)
sample_individual_estimates(model, samples_per_id=2, rng=rng)

## End(Not run)
```

```
sample_parameters_from_covariance_matrix
      sample_parameters_from_covariance_matrix
```

Description

Sample parameter vectors using the covariance matrix

If modelfit_results is not provided the results from the model will be used

Usage

```
sample_parameters_from_covariance_matrix(
  model,
  modelfit_results = NULL,
  parameters = NULL,
  force_posdef_samples = NULL,
  force_posdef_covmatrix = FALSE,
  n = 1,
  rng = NULL
)
```

Arguments

model	(Model) Input model
modelfit_results	(ModelfitResults) Alternative results object. Default is to use the one in model
parameters	(vector) Use to only sample a subset of the parameters. NULL means all
force_posdef_samples	(integer) Set to how many iterations to do before forcing all samples to be positive definite. NULL is default and means never and 0 means always
force_posdef_covmatrix	(logical) Set to TRUE to force the input covariance matrix to be positive definite
n	(integer) Number of samples
rng	(Generator) Random number generator

Value

(data.frame) A dataframe with one sample per row

See Also

sample_parameters_uniformly : Sample parameter vectors using uniform distribution
 sample_individual_estimates : Sample individual estimates given their covariance

Examples

```
## Not run:
model <- load_example_model("pheno")
rng <- create_rng(23)
sample_parameters_from_covariance_matrix(model, n=3, rng=rng)

## End(Not run)
```

```
sample_parameters_uniformly
      sample_parameters_uniformly
```

Description

Sample parameter vectors using uniform sampling

Each parameter value will be randomly sampled from a uniform distribution with the bounds being estimate \pm estimate * fraction.

Usage

```
sample_parameters_uniformly(
  model,
  fraction = 0.1,
  parameters = NULL,
  force_posdef_samples = NULL,
  n = 1,
  rng = NULL
)
```

Arguments

model	(Model) Pharmpy model
fraction	(numeric) Fraction of estimate value to use for distribution bounds
parameters	(data.frame) Names of parameters to use. Default is to use all parameters in the model.
force_posdef_samples	(integer) Number of samples to reject before forcing variability parameters to give positive definite covariance matrices.
n	(integer) Number of samples
rng	(integer or rng) Random number generator or seed

Value

(data.frame) samples

See Also

sample_parameters_from_covariance_matrix : Sample parameter vectors using the uncertainty covariance matrix

sample_individual_estimates : Sample individual estimates given their covariance

Examples

```
## Not run:
model <- load_example_model("pheno")
rng <- create_rng(23)
sample_parameters_uniformly(model, n=3, rng=rng)

## End(Not run)
```

```
set_additive_error_model
      set_additive_error_model
```

Description

Set an additive error model. Initial estimate for new sigma is $(\min(DV)/2)^2$.

The error function being applied depends on the data transformation. The table displays some examples.

	Data transformation	Additive error
$f + \epsilon_1$	$\log(y)$	y
$\log(f) + \frac{\epsilon_1}{f}$		

Usage

```
set_additive_error_model(model, data_trans = NULL, series_terms = 2)
```

Arguments

model	(Model) Set error model for this model
data_trans	(str or expression) A data transformation expression or NULL (default) to use the transformation specified by the model. Series expansion will be used for approximation.
series_terms	(integer) Number of terms to use for the series expansion approximation for data transformation.

Value

(Model) Reference to the same model object

See Also

set_proportional_error_model : Proportional error model

set_combined_error_model : Combined error model

Examples

```
## Not run:
model <- load_example_model("pheno")
model$statements$find_assignment("Y")
set_additive_error_model(model)
model$statements$find_assignment("Y")
model <- load_example_model("pheno")
model$statements$find_assignment("Y")
set_additive_error_model(model, data_trans="log(Y)")
model$statements$find_assignment("Y")

## End(Not run)
```

set_bolus_absorption *set_bolus_absorption*

Description

Set or change to bolus absorption rate.

Usage

```
set_bolus_absorption(model)
```

Arguments

model (Model) Model to set or change absorption rate

Value

(Model) Reference to same model

See Also

set_zero_order_absorption

set_first_order_absorption

Examples

```
## Not run:
model <- remove_error_model(load_example_model("pheno"))
set_combined_error_model(model)
model$statements$find_assignment("Y")
model <- remove_error_model(load_example_model("pheno"))
set_combined_error_model(model, data_trans="log(Y)")
model$statements$find_assignment("Y")

## End(Not run)
```

set_dtbs_error_model *set_dtbs_error_model*

Description

Dynamic transform both sides

Usage

```
set_dtbs_error_model(model)
```

Arguments

model (Model) PharmPy model

Value

(Model) Reference to the same model

Examples

```
## Not run:
model <- load_example_model("pheno")
set_dtbs_error_model(model)

## End(Not run)
```

set_estimation_step *set_estimation_step*

Description

Set estimation step

Sets estimation step for a model. Methods currently supported are: FO, FOCE, ITS, LAPLACE, IMPMAP, IMP, SAEM, BAYES

Usage

```
set_estimation_step(model, method, interaction = TRUE, options = NULL, idx = 0)
```

Arguments

model	(Model) PharmPy model
method	(str) estimation method to change to
interaction	(logical) whether to use interaction or not, default is true
options	(list) any additional options. Note that this replaces old options (see <code>append_estimation_step_options</code> to keep old options)
idx	(integer) index of estimation step, default is 0 (first estimation step)

Value

(Model) Reference to the same model object

See Also

`add_estimation_step`
`remove_estimation_step`
`append_estimation_step_options`

Examples

```
## Not run:  
model <- load_example_model("pheno")  
opts <- list('NITER'=1000, 'ISAMPLE'=100, 'EONLY'=1)  
set_estimation_step(model, "IMP", options=opts)  
model$estimation_steps[1]  
  
## End(Not run)
```

```
set_first_order_absorption  
    set_first_order_absorption
```

Description

Set or change to first order absorption rate.

Initial estimate for absorption rate is set to the previous rate if available, otherwise it is set to the time of first observation/2.

Usage

```
set_first_order_absorption(model)
```

Arguments

model (Model) Model to set or change to use first order absorption rate

Value

(Model) Reference to same model

See Also

set_bolus_order_absorption

set_zero_order_absorption

Examples

```
## Not run:  
model <- load_example_model("pheno")  
set_first_order_absorption(model)  
model$statements$ode_system  
  
## End(Not run)
```

```
set_first_order_elimination  
    set_first_order_elimination
```

Description

Sets elimination to first order

Usage

```
set_first_order_elimination(model)
```

Arguments

model (Model) PharmPy model

Value

(Model) Reference to same model

See Also

```
set_zero_order_elimination
set_michaelis_menten_elimination
```

Examples

```
## Not run:
model <- load_example_model("pheno")
set_first_order_elimination(model)
model$statements$ode_system

## End(Not run)
```

```
set_iiv_on_ruv      set_iiv_on_ruv
```

Description

Multiplies epsilons with exponential (new) etas.
Initial variance for new etas is 0.09.

Usage

```
set_iiv_on_ruv(model, list_of_eps = NULL, same_eta = TRUE, eta_names = NULL)
```

Arguments

model (Model) PharmPy model to apply IIV on epsilons.

list_of_eps (str, vector) Name/names of epsilons to multiply with exponential etas. If NULL, all epsilons will be chosen. NULL is default.

same_eta (logical) Boolean of whether all RUVs from input should use the same new ETA or if one ETA should be created for each RUV. TRUE is default.

eta_names (str, vector) Custom names of new etas. Must be equal to the number epsilons or 1 if same eta.

Value

(Model) Reference to same model

See Also

set_power_on_ruv

Examples

```
## Not run:  
model <- load_example_model("pheno")  
set_iiv_on_ruv(model)  
model$statements$find_assignment("Y")  
  
## End(Not run)
```

set_initial_estimates *set_initial_estimates*

Description

Set initial estimates

Usage

```
set_initial_estimates(model, inits)
```

Arguments

model (Model) PharmPy model
inits (list) A list of parameter init for parameters to change

Value

(Model) Reference to the same model object

Examples

```
## Not run:  
model <- load_example_model("pheno")  
set_initial_estimates(model, list('THETA(1)'=2))  
model$parameters['THETA(1)']  
  
## End(Not run)
```

set_lag_time	<i>set_lag_time</i>
--------------	---------------------

Description

Add lag time to the dose compartment of model.

Initial estimate for lag time is set the previous lag time if available, otherwise it is set to the time of first observation/2.

Usage

```
set_lag_time(model)
```

Arguments

model (Model) Pharnpy model

Value

(Model) Reference to same model

See Also

set_transit_compartments

remove_lag_time

Examples

```
## Not run:  
model <- load_example_model("pheno")  
set_lag_time(model)  
  
## End(Not run)
```

set_michaelis_menten_elimination	<i>set_michaelis_menten_elimination</i>
----------------------------------	---

Description

Sets elimination to Michaelis-Menten.

Initial estimate for CLMM is set to CL and KM is set to $2 \times \max(DV)$.

Usage

```
set_michaelis_menten_elimination(model)
```

Arguments

model (Model) PharmPy model

Value

(Model) Reference to the same model

See Also

set_first_order_elimination
set_zero_order_elimination

Examples

```
## Not run:  
model <- load_example_model("pheno")  
set_michaelis_menten_elimination(model)  
model$statements$ode_system  
  
## End(Not run)
```

```
set_mixed_mm_fo_elimination  
    set_mixed_mm_fo_elimination
```

Description

Sets elimination to mixed Michaelis-Menten and first order.

Initial estimate for CLMM is set to $CL/2$ and KM is set to $2 \cdot \max(DV)$.

Usage

```
set_mixed_mm_fo_elimination(model)
```

Arguments

model (Model) PharmPy model

Value

(Model) Reference to the same model

See Also

set_first_order_elimination
set_zero_order_elimination
set_michaelis_menten_elimination

Examples

```
## Not run:
model <- load_example_model("pheno")
set_mixed_mm_fo_elimination(model)
model$statements$ode_system

## End(Not run)
```

set_name	<i>set_name</i>
----------	-----------------

Description

Set name of model object

Usage

```
set_name(model, new_name)
```

Arguments

model	(Model) Pharmpy model
new_name	(str) New name of model

Value

(Model) Reference to the same model object

Examples

```
## Not run:
model <- load_example_model("pheno")
model$name
set_name(model, "run2")
model$name

## End(Not run)
```

```
set_ode_solver      set_ode_solver
```

Description

Sets ODE solver to use for model

Recognized solvers and their corresponding NONMEM advans:

```
+-----+-----+ | Solver | NONMEM ADVAN | +=====+=====+
| CVODES | ADVAN14 | +-----+-----+ | DGEAR | ADVAN8 | +-----
-----+-----+ | DVERK | ADVAN6 | +-----+-----+ | IDA
| ADVAN15 | +-----+-----+ | LSODA | ADVAN13 | +-----
+-----+-----+ | LSODI | ADVAN9 | +-----+-----+
```

Usage

```
set_ode_solver(model, solver)
```

Arguments

```
model      (Model) Pharnpy model
solver     (str) Solver to use or NULL for no preference
```

Value

(Model) Reference to same model

Examples

```
## Not run:
model <- load_example_model("pheno")
set_ode_solver(model, 'LSODA')

## End(Not run)
```

```
set_peripheral_compartments
      set_peripheral_compartments
```

Description

Sets the number of peripheral compartments to a specified number.

Usage

```
set_peripheral_compartments(model, n)
```

Arguments

`model` (Model) PharmPy model
`n` (integer) Number of transit compartments

Value

(Model) Reference to same model

See Also

`add_peripheral_compartment`
`remove_peripheral_compartment`

Examples

```
## Not run:
model <- load_example_model("pheno")
set_peripheral_compartments(model, 2)
model$statements$ode_system

## End(Not run)
```

`set_power_on_ruv` *set_power_on_ruv*

Description

Applies a power effect to provided epsilons.

Initial estimates for new thetas are 1 if the error model is proportional, otherwise they are 0.1.

Usage

```
set_power_on_ruv(model, list_of_eps = NULL, ipred = NULL)
```

Arguments

`model` (Model) PharmPy model to create block effect on.
`list_of_eps` (str, vector) Name/names of epsilons to apply power effect. If NULL, all epsilons will be used. NULL is default.
`ipred` (Symbol) Symbol to use as IPRED. Default is to autodetect expression for IPRED.

Value

(Model) Reference to the same model

See Also

set_iiv_on_ruv

Examples

```
## Not run:
model <- load_example_model("pheno")
set_power_on_ruv(model)
model$statements$find_assignment("Y")

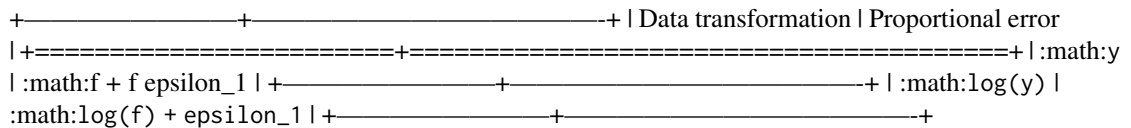
## End(Not run)
```

set_proportional_error_model
set_proportional_error_model

Description

Set a proportional error model. Initial estimate for new sigma is 0.09.

The error function being applied depends on the data transformation.



Usage

```
set_proportional_error_model(model, data_trans = NULL)
```

Arguments

- model (Model) Set error model for this model
- data_trans (str or expression) A data transformation expression or NULL (default) to use the transformation specified by the model.

Value

(Model) Reference to the same model object

See Also

- set_additive_error_model : Additive error model
- set_combined_error_model : Combined error model

Examples

```
## Not run:
model <- remove_error_model(load_example_model("pheno"))
set_proportional_error_model(model)
model$statements$find_assignment("Y")
model <- remove_error_model(load_example_model("pheno"))
set_proportional_error_model(model, data_trans="log(Y)")
model$statements$find_assignment("Y")

## End(Not run)
```

```
set_seq_zo_fo_absorption
      set_seq_zo_fo_absorption
```

Description

Set or change to sequential zero order first order absorption rate.

Initial estimate for absorption rate is set the previous rate if available, otherwise it is set to the time of first observation/2.

Usage

```
set_seq_zo_fo_absorption(model)
```

Arguments

model (Model) Model to set or change absorption rate

Value

(Model) Reference to same model

See Also

```
set_bolus_order_absorption
set_zero_order_absorption
set_first_order_absorption
```

Examples

```
## Not run:
model <- load_example_model("pheno")
set_seq_zo_fo_absorption(model)
model$statements$ode_system

## End(Not run)
```

```
set_transit_compartments  
    set_transit_compartments
```

Description

Set the number of transit compartments of model.

Initial estimate for absorption rate is set the previous rate if available, otherwise it is set to the time of first observation/2.

Usage

```
set_transit_compartments(model, n)
```

Arguments

model	(Model) PharmPy model
n	(integer) Number of transit compartments

Value

(Model) Reference to same model

See Also

set_lag_time

Examples

```
## Not run:  
model <- load_example_model("pheno")  
set_transit_compartments(model, 3)  
model$statements$ode_system  
  
## End(Not run)
```

```
set_weighted_error_model  
    set_weighted_error_model
```

Description

Encode error model with one epsilon and W as weight

Usage

```
set_weighted_error_model(model)
```

Arguments

```
model          (Model) Pharmpy model
```

Value

(Model) Reference to the same model

See Also

use_thetas_for_error_stdev : Use thetas to estimate error

Examples

```
## Not run:  
model <- load_example_model("pheno")  
set_weighted_error_model(model)  
  
## End(Not run)
```

```
set_zero_order_absorption  
    set_zero_order_absorption
```

Description

Set or change to zero order absorption rate.

Initial estimate for absorption rate is set the previous rate if available, otherwise it is set to the time of first observation/2.

Usage

```
set_zero_order_absorption(model)
```

Arguments

```
model          (Model) Model to set or change to first order absorption rate
```

Value

(Model) Reference to the same model

See Also

```
set_bolus_order_absorption  
set_first_order_absorption
```

Examples

```
## Not run:  
model <- load_example_model("pheno")  
set_zero_order_absorption(model)  
model$statements$ode_system  
  
## End(Not run)
```

```
set_zero_order_elimination  
    set_zero_order_elimination
```

Description

Sets elimination to zero order.
Initial estimate for KM is set to 1% of smallest observation.

Usage

```
set_zero_order_elimination(model)
```

Arguments

model (Model) PharmPy model

Value

(Model) Reference to same model

See Also

set_first_order_elimination
set_michaelis_menten_elimination

Examples

```
## Not run:  
model <- load_example_model("pheno")  
set_zero_order_elimination(model)  
model$statements$ode_system  
  
## End(Not run)
```

split_joint_distribution
split_joint_distribution

Description

Splits etas following a joint distribution into separate distributions.

Usage

```
split_joint_distribution(model, rvs = NULL)
```

Arguments

model	(Model) PharmPy model
rvs	(str, vector) Name/names of etas to separate. If NULL, all etas that are IIVs and non-fixed will become single. NULL is default.

Value

(Model) Reference to the same model

See Also

create_joint_distribution : combine etas into a join distribution

Examples

```
## Not run:  
model <- load_example_model("pheno")  
create_joint_distribution(model, c('ETA(1)', 'ETA(2)'))  
model$random_variables$etas  
split_joint_distribution(model, c('ETA(1)', 'ETA(2)'))  
model$random_variables$etas  
  
## End(Not run)
```

summarize_modelfit_results
summarize_modelfit_results

Description

Summarize results of model runs

Summarize different results after fitting a model, includes runtime, ofv, and parameter estimates (with errors).

Usage

```
summarize_modelfit_results(models)
```

Arguments

models (vector, Model) List of models or single model

Value

(data.frame) A DataFrame of modelfit results, one row per model.

Examples

```
## Not run:  
model <- load_example_model("pheno")  
summarize_modelfit_results(c(model))  
  
## End(Not run)
```

transform_etas_boxcox *transform_etas_boxcox*

Description

Applies a boxcox transformation to selected etas
Initial estimate for lambda is 0.1 with bounds (-3, 3).

Usage

```
transform_etas_boxcox(model, list_of_etas = NULL)
```

Arguments

model (Model) PharmPy model to apply boxcox transformation to.
list_of_etas (str, vector) Name/names of etas to transform. If NULL, all etas will be transformed (default).

Value

(Model) Reference to the same model

See Also

transform_etas_tdist
transform_etas_john_draper

Examples

```
## Not run:
model <- load_example_model("pheno")
transform_etas_boxcox(model, c("ETA(1)"))
model$statements$full_expression_from_odes("CL")

## End(Not run)
```

```
transform_etas_john_draper
      transform_etas_john_draper
```

Description

Applies a John Draper transformation (1) to spelected etas

Initial estimate for lambda is 0.1 with bounds (-3, 3).

(1) John, J., Draper, N. (1980). An Alternative Family of Transformations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29(2), 190-197. doi:10.2307/2986305

Usage

```
transform_etas_john_draper(model, list_of_etas = NULL)
```

Arguments

model	(Model) Pharmpy model to apply John Draper transformation to.
list_of_etas	(str, vector) Name/names of etas to transform. If NULL, all etas will be transformed (default).

Value

(Model) Reference to the same model

See Also

```
transform_etas_boxcox
transform_etas_tdist
```

Examples

```
## Not run:
model <- load_example_model("pheno")
transform_etas_john_draper(model, c("ETA(1)"))
model$statements$full_expression_from_odes("CL")

## End(Not run)
```

transform_etas_tdist *transform_etas_tdist*

Description

Applies a t-distribution transformation to selected etas

Initial estimate for degrees of freedom is 80 with bounds (3, 100).

Usage

```
transform_etas_tdist(model, list_of_etas = NULL)
```

Arguments

`model` (Model) PharmPy model to apply t distribution transformation to.

`list_of_etas` (str, vector) Name/names of etas to transform. If NULL, all etas will be transformed (default).

Value

(Model) Reference to the same model

See Also

`transform_etas_boxcox`

`transform_etas_john_draper`

Examples

```
## Not run:  
model <- load_example_model("pheno")  
transform_etas_tdist(model, c("ETA(1)"))  
model$statements$full_expression_from_odes("CL")  
  
## End(Not run)
```

unfix_parameters	<i>unfix_parameters</i>
------------------	-------------------------

Description

Unfix parameters

Unfix all listed parameters

Usage

```
unfix_parameters(model, parameter_names)
```

Arguments

model (Model) Pharmpy model

parameter_names
(vector or str) one parameter name or a vector of parameter names

Value

(Model) Reference to the same model object

See Also

unfix_paramaters_to : Unfixing parameters and setting a new initial estimate in the same function

fix_parameters : Fix parameters

fix_parameters_to : Fixing and setting parameter initial estimates in the same function

Examples

```
## Not run:  
model <- load_example_model("pheno")  
fix_parameters(model, c('THETA(1)', 'THETA(2)', 'THETA(3)'))  
model$parameters$fix  
unfix_parameters(model, 'THETA(1)')  
model$parameters$fix  
  
## End(Not run)
```

unfix_parameters_to *unfix_parameters_to*

Description

Unix parameters to
 Unfix all listed parameters to specified value/values

Usage

```
unfix_parameters_to(model, parameter_names, values)
```

Arguments

model	(Model) Pharmpy model
parameter_names	(vector or str) one parameter name or a vector of parameter names
values	(vector or numeric) one value or a vector of values (must be equal to number of parameter_names)

Value

(Model) Reference to the same model object

Examples

```
## Not run:
model <- load_example_model("pheno")
fix_parameters(model, c('THETA(1)', 'THETA(2)', 'THETA(3)'))
model$parameters$fix
unfix_parameters_to(model, 'THETA(1)', 0.5)
model$parameters$fix
model$parameters['THETA(1)']

## End(Not run)
```

update_inits *update_inits*

Description

Update initial parameter estimate for a model
 Updates initial estimates of population parameters for a model from its modelfit_results. If the model has used initial estimates for individual estimates these will also be updated. If initial estimates

Usage

```
update_inits(model, force_individual_estimates = FALSE)
```

Arguments

`model` (Model) Pharmpy model to update initial estimates

`force_individual_estimates` (logical) Update initial individual estimates even if model didn't use them previously.

Value

(Model) Reference to the same model

Examples

```
## Not run:
model <- load_example_model("pheno")
model$parameters$inits
update_inits(model)
model$parameters$inits

## End(Not run)
```

update_source

update_source

Description

Update source

Let the code of the underlying source language be updated to reflect changes in the model object.

Usage

```
update_source(model)
```

Arguments

`model` (Model) Pharmpy model

Value

(Model) Reference to the same model object

Examples

```
## Not run:
model <- load_example_model("pheno")
fix_parameters(model, c('THETA(1)'))
update_source(model)
print(str(model).splitlines()c(22))

## End(Not run)
```

```
use_thetas_for_error_stdev
      use_thetas_for_error_stdev
```

Description

Use thetas to estimate standard deviation of error

Usage

```
use_thetas_for_error_stdev(model)
```

Arguments

model (Model) PharmPy model

Value

(Model) Reference to the same model

See Also

set_weighted_error_model : Encode error model with one epsilon and weight

Examples

```
## Not run:
model <- load_example_model("pheno")
use_thetas_for_error_stdev(model)
model$statements$find_assignment("Y")

## End(Not run)
```

`write_model`*write_model*

Description

Write model code to file

Usage

```
write_model(model, path = "", force = TRUE)
```

Arguments

<code>model</code>	(Model) Pharmpy model
<code>path</code>	(str) Destination path
<code>force</code>	(logical) Force overwrite, default is TRUE

Value

(Model) Reference to the same model object

Examples

```
## Not run:  
model <- load_example_model("pheno")  
write_model(model)  
  
## End(Not run)
```


Index

add_covariate_effect, 4
add_estimation_step, 6
add_iiv, 7
add_individual_parameter, 9
add_iov, 9
add_peripheral_compartment, 10
append_estimation_step_options, 11

calculate_eta_shrinkage, 12
calculate_individual_parameter_statistics, 13
calculate_individual_shrinkage, 14
calculate_pk_parameters_statistics, 14
check_pharmpy, 15
convert_model, 16
copy_model, 16
create_joint_distribution, 17
create_results, 18
create_rng, 19

evaluate_expression, 19

fit, 20
fix_parameters, 21
fix_parameters_to, 22

get_model_covariates, 23
get_number_of_individuals, 23
get_number_of_observations, 24
get_number_of_observations_per_individual, 25
get_observations, 26

has_additive_error_model, 27
has_combined_error_model, 27
has_proportional_error_model, 28
has_zero_order_absorption, 29

install_pharmpy, 30
load_example_model, 30

plot_individual_predictions, 31
plot_iofv_vs_iofv, 31
predict_influential_individuals, 32
predict_influential_outliers, 32
predict_outliers, 33
print_model_symbols, 34

read_model, 34
read_model_from_database, 35
read_model_from_string, 36
read_results, 37
remove_error_model, 37
remove_estimation_step, 38
remove_iiv, 39
remove_iov, 40
remove_lag_time, 40
remove_peripheral_compartment, 41
reset_index, 42
run_tool, 42

sample_individual_estimates, 43
sample_parameters_from_covariance_matrix, 44
sample_parameters_uniformly, 45
set_additive_error_model, 46
set_bolus_absorption, 47
set_combined_error_model, 48
set_dtbs_error_model, 49
set_estimation_step, 50
set_first_order_absorption, 51
set_first_order_elimination, 51
set_iiv_on_ruv, 52
set_initial_estimates, 53
set_lag_time, 54
set_michaelis_menten_elimination, 54
set_mixed_mm_fo_elimination, 55
set_name, 56
set_ode_solver, 57
set_peripheral_compartments, 57
set_power_on_ruv, 58

set_proportional_error_model, [59](#)
set_seq_zo_fo_absorption, [60](#)
set_transit_compartments, [61](#)
set_weighted_error_model, [61](#)
set_zero_order_absorption, [62](#)
set_zero_order_elimination, [63](#)
split_joint_distribution, [64](#)
summarize_modelfit_results, [64](#)

transform_etas_boxcox, [65](#)
transform_etas_john_draper, [66](#)
transform_etas_tdist, [67](#)

unfix_parameters, [68](#)
unfix_parameters_to, [69](#)
update_inits, [69](#)
update_source, [70](#)
use_thetas_for_error_stdev, [71](#)

write_model, [72](#)