Package ‘pmxTools’

January 10, 2020

Type Package

Title Pharmacometric and Pharmacokinetic Toolkit

Version 1.0

Maintainer Justin Wilkins <justin.wilkins@occams.com>

Contact Justin Wilkins <justin.wilkins@occams.com>

Description Pharmacometric tools for common data analytical tasks; closed-form solutions for calculating concentrations at given times after dosing based on compartmental PK models (1-compartment, 2-compartment and 3-compartment, covering infusions, zero- and first-order absorption, and lag times, after single doses and at steady state, per Bertrand & Mentre (2008) <http://lixoft.com/wp-content/uploads/2016/03/PKPDlibrary.pdf>); parametric simulation from NONMEM-generated parameter estimates and other output; and parsing, tabulating and plotting results generated by Perl-speaks-NONMEM (PsN).

License GPL-2

URL https://github.com/kestrel99/pmxTools

BugReports https://github.com/kestrel99/pmxTools/issues

LazyData TRUE

RoxygenNote 7.0.2

Imports ggplot2, ggrepel, gridExtra, chron

Depends MASS, stringr, XML

Suggests testthat

Encoding UTF-8

NeedsCompilation no

Author Justin Wilkins [aut, cre] (<https://orcid.org/0000-0002-7099-9396>),
Bill Denney [aut] (<https://orcid.org/0000-0002-5759-428X>),
Rik Schoemaker [aut],
Leonid Gibiansky [ctb],

Andrew Hooker [ctb],
E. Niclas Jonsson [ctb],
Mats O. Karlsson [ctb],
John Johnson [ctb]

Repository CRAN

Date/Publication 2020-01-10 07:50:02 UTC

R topics documented:

`calc_derived` *Calculate derived pharmacokinetic parameters for a 1-, 2-, or 3-compartment linear model.*

Description

Calculate derived pharmacokinetic parameters for a 1-, 2-, or 3-compartment linear model.

Usage

```
calc_derived(..., verbose = FALSE)

calc_derived_1cpt(
  CL,
  V = NULL,
  V1 = NULL,
  ka = NULL,
  tlag = NULL,
  type = "all",
  sigdig = 5
)

calc_derived_2cpt(
  CL,
  V1 = NULL,
  V2,
  Q2 = NULL,
  V = NULL,
  Q = NULL,
  ka = NULL,
  tlag = NULL,
  type = "all",
  sigdig = 5
)

calc_derived_3cpt(
  CL,
  V1 = NULL,
  V2,
  V3,
  Q2 = NULL,
  Q3,
  V = NULL,
  Q = NULL,
  ka = NULL,
  tlag = NULL,
  type = "all",
  sigdig = 5
)
```

Arguments

`...` Passed to the other ‘calc_derived_*()‘ functions.

`verbose` For ‘calc_derived()‘, provide a message indicating the type of model detected.

`CL` Clearance (L/h)

`V1, V` Central volume of distribution (L). Values are synonyms; use only one.

`ka` Absorption rate (1/hr)

`tlag` Absorption lag time (hr)

`type` Parameters to return. Default is `"all"`; see details for other options that are specific to the number of compartments.

`sigdig` Number of significant digits to be returned. Default is `5`.

`V2` First peripheral volume of distribution (L)

`Q2, Q` Intercompartmental clearance from central to first peripheral compartment (L/h). Values are synonyms; use only one.

`V3` Second peripheral volume of distribution (L)

`Q3` Intercompartmental clearance from central to second peripheral compartment (L/h)

Value

Return a list of derived PK parameters for a 1-, 2-, or 3-compartment linear model given provided clearances and volumes based on the ‘type‘.

- `Vss`: Volume of distribution at steady state, V_{ss} (L); 1-, 2-, and 3-compartment
- `k10`: First-order elimination rate, k_{10} (/h); 1-, 2-, and 3-compartment
- `k12`: First-order rate of transfer from central to first peripheral compartment, k_{12} (/h); 2- and 3-compartment
- `k21`: First-order rate of transfer from first peripheral to central compartment, k_{21} (/h); 2- and 3-compartment
- `k13`: First-order rate of transfer from central to second peripheral compartment, k_{13} (/h); 3-compartment
- `k31`: First-order rate of transfer from second peripheral to central compartment,k_{31} (/h); 3-compartment
- `thalf_alpha`: $t_{1/2,\alpha}$ (h); 1-, 2-, and 3-compartment
- `thalf_beta`: $t_{1/2,\beta}$ (h); 2- and 3-compartment
- `thalf_gamma`: $t_{1/2,\gamma}$ (h); 3-compartment
- `alpha`: α; 1-, 2-, and 3-compartment
- `beta`: β; 2- and 3-compartment
- `gamma`: β; 3-compartment

- `trueA`: true A; 1-, 2-, and 3-compartment
- `trueB`: true B; 2- and 3-compartment
- `trueC`: true C; 3-compartment
- `fracA`: fractional A; 1-, 2-, and 3-compartment
- `fracB`: fractional B; 2- and 3-compartment
- `fracC`: fractional C; 3-compartment

The input parameters with standardized names (‘V1‘, ‘V2‘, ‘V3‘, ‘CL‘, ‘Q2‘, and ‘Q3‘) are also returned in the list, and if provided, additional PK parameters of ‘ka‘ and ‘lag‘ are also returned in the list. All inputs may be scalars or vectors.

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

References

Shafer S. L. `CONVERT.XLS`

Rowland M, Tozer TN. Clinical Pharmacokinetics and Pharmacodynamics: Concepts and Applications (4th). Lippincott Williams & Wilkins, Philadelphia, 2010.

Examples

```
params <- calc_derived(CL=29.4, V1=23.4, V2=114, V3=4614, Q2=270, Q3=73)
params <- calc_derived_1cpt(CL=16, V=25)
params <- calc_derived_2cpt(CL=16, V1=25, V2=50, Q=0.5)
params <- calc_derived_3cpt(CL=29.4, V1=23.4, V2=114, V3=4614, Q2=270, Q3=73)
```

`calc_sd_1cmt` *Calculate C(t) for a 1-compartment linear model*

Description

Calculate C(t) for a 1-compartment linear model

Usage

```
calc_sd_1cmt(t, dose, dur = NULL, tinf = NULL, ...)

calc_sd_1cmt_linear_bolus(t, dose, ...)

calc_sd_1cmt_linear_oral_1_lag(t, dose, ...)

calc_sd_1cmt_linear_infusion(t, dose, tinf, ...)

calc_sd_1cmt_linear_oral_0(t, dose, dur, ...)
```

```
calc_sd_1cmt_linear_oral_1(t, dose, ...)

calc_sd_1cmt_linear_oral_0_lag(t, dose, dur, ...)
```

Arguments

`t`	Time after dose (h)
`dose`	Dose
`dur`	Duration of zero-order absorption (h)
`tinf`	Duration of infusion (h)
`...`	Passed to ‘calc_derived_1cpt()‘

Value

Concentration of drug at requested time (`t`) after a single dose, given provided set of parameters and variables.

Functions

- `calc_sd_1cmt_linear_bolus`: Calculate C(t) for a 1-compartment linear model after a single IV bolus dose
- `calc_sd_1cmt_linear_oral_1_lag`: Calculate C(t) for a 1-compartment linear model with first-order absorption after a single oral dose, with lag time
- `calc_sd_1cmt_linear_infusion`: Calculate C(t) for a 1-compartment linear model after a single IV infusion
- `calc_sd_1cmt_linear_oral_0`: Calculate C(t) for a 1-compartment linear model with zero-order absorption after a single oral dose
- `calc_sd_1cmt_linear_oral_1`: Calculate C(t) for a 1-compartment linear model with first-order absorption after a single oral dose
- `calc_sd_1cmt_linear_oral_0_lag`: Calculate C(t) for a 1-compartment linear model with zero-order absorption after a single oral dose, with lag time

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

References

Bertrand J & Mentre F (2008). Mathematical Expressions of the Pharmacokinetic and Pharmacodynamic Models implemented in the Monolix software. http://lixoft.com/wp-content/uploads/2016/03/PKPDlibrary.pdf

Rowland M, Tozer TN. Clinical Pharmacokinetics and Pharmacodynamics: Concepts and Applications (4th). Lippincott Williams & Wilkins, Philadelphia, 2010.

Examples

```
Ct <- calc_sd_1cmt_linear_bolus(t=0:24, CL=6, V=25, dose=600)
Ct <- calc_sd_1cmt_linear_oral_1_lag(t=0:24, CL=6, V=25, ka=1.1, dose=600, tlag=2)
Ct <- calc_sd_1cmt_linear_infusion(t=0:24, CL=6, V=25, dose=600, tinf=1)
Ct <- calc_sd_1cmt_linear_oral_0(t=0:24, CL=6, V=25, dur=1.5, dose=600)
Ct <- calc_sd_1cmt_linear_oral_1(t=0:24, CL=6, V=25, ka=1.1, dose=600)
Ct <- calc_sd_1cmt_linear_oral_0_lag(t=0:24, CL=6, V=25, dur=1.5, dose=600, tlag=1.5)
```

calc_sd_2cmt *Calculate C(t) for a 1-compartment linear model*

Description

Calculate C(t) for a 1-compartment linear model

Usage

```
calc_sd_2cmt(t, dose, dur = NULL, tinf = NULL, ...)

calc_sd_2cmt_linear_bolus(t, dose, ...)

calc_sd_2cmt_linear_oral_1_lag(t, dose, ...)

calc_sd_2cmt_linear_infusion(t, dose, tinf, ...)

calc_sd_2cmt_linear_oral_0_lag(t, dose, dur, ...)

calc_sd_2cmt_linear_oral_0_lag(t, dose, dur, ...)

calc_sd_2cmt_linear_oral_1(t, dose, ...)

calc_sd_2cmt_linear_oral_0(t, dose, dur, ...)
```

Arguments

`t`	Time after dose (h)
`dose`	Dose
`dur`	Duration of zero-order absorption (h)
`tinf`	Duration of infusion (h)
`...`	Passed to ‘calc_derived_2cpt()‘

Value

Concentration of drug at requested time (`t`) after a single dose, given provided set of parameters and variables.

Functions

- `calc_sd_2cmt_linear_bolus`: Calculate C(t) for a 2-compartment linear model after a single IV bolus dose
- `calc_sd_2cmt_linear_oral_1_lag`: Calculate C(t) for a 2-compartment linear model after a single first-order oral dose with a lag time
- `calc_sd_2cmt_linear_infusion`: Calculate C(t) for a 2-compartment linear model after a single infusion
- `calc_sd_2cmt_linear_oral_0_lag`: Calculate C(t) for a 2-compartment linear model after a single zero-order oral dose, with lag time
- `calc_sd_2cmt_linear_oral_0_lag`: Calculate C(t) for a 2-compartment linear model after a single zero-order oral dose, with lag time
- `calc_sd_2cmt_linear_oral_1`: Calculate C(t) for a 2-compartment linear model after a single first-order oral dose
- `calc_sd_2cmt_linear_oral_0`: Calculate C(t) for a 2-compartment linear model after a single zero-order oral dose

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

References

Bertrand J & Mentre F (2008). Mathematical Expressions of the Pharmacokinetic and Pharmacodynamic Models implemented in the Monolix software. http://lixoft.com/wp-content/uploads/2016/03/PKPDlibrary.pdf

Rowland M, Tozer TN. Clinical Pharmacokinetics and Pharmacodynamics: Concepts and Applications (4th). Lippincott Williams & Wilkins, Philadelphia, 2010.

Examples

```
Ct <- calc_sd_2cmt_linear_bolus(t = 11.75, CL = 7.5, V1 = 20, V2 = 30, Q = 0.5, dose = 10)
Ct <- calc_sd_2cmt_linear_oral_1_lag(t = 11.75, CL = 7.5, V1 = 20, V2 = 30, Q = 0.5,
    dose = 1000, ka = 1, tlag = 2)
Ctrough <- calc_sd_2cmt_linear_infusion(t = 11.75, CL = 7.5, V1 = 20, V2 = 30, Q = 0.5,
    dose = 10, tinf = 1)
Ctrough <- calc_sd_2cmt_linear_oral_0_lag(t = 11.75, CL = 7.5, V1 = 20, V2 = 30, Q = 0.5,
    dose = 1000, dur = 1, tlag=2)
Ct <- calc_sd_2cmt_linear_oral_0_lag(t = 11.75, CL = 7.5, V1 = 20, V2 = 30, Q = 0.5,
    dose = 1000, dur = 1, tlag=2)
Ct <- calc_sd_2cmt_linear_oral_1(t = 11.75, CL = 7.5, V1 = 20, V2 = 30, Q = 0.5,
    dose = 1000, ka = 1)
Ct <- calc_sd_2cmt_linear_oral_0(t = 11.75, CL = 7.5, V1 = 20, V2 = 30, Q = 0.5,
    dose = 1000, dur = 1)
```

calc_sd_3cmt *Calculate C(t) for a 1-compartment linear model*

Description

Calculate C(t) for a 1-compartment linear model

Usage

```
calc_sd_3cmt(t, dose, dur = NULL, tinf = NULL, ...)

calc_sd_3cmt_linear_bolus(t, dose, ...)

calc_sd_3cmt_linear_oral_1_lag(t, dose, ...)

calc_sd_3cmt_linear_infusion(t, dose, tinf, ...)

calc_sd_3cmt_linear_oral_0(t, dose, dur, ...)

calc_sd_3cmt_linear_oral_0_lag(t, dose, dur, ...)

calc_sd_3cmt_linear_oral_1(t, dose, ...)
```

Arguments

`t`	Time after dose (h)
`dose`	Dose
`dur`	Duration of zero-order absorption (h)
`tinf`	Duration of infusion (h)
`...`	Passed to ‘calc_derived_3cpt()‘

Value

Concentration of drug at requested time (`t`) after a single dose, given provided set of parameters and variables.

Functions

- `calc_sd_3cmt_linear_bolus`: Calculate C(t) for a 3-compartment linear model after a single IV bolus dose
- `calc_sd_3cmt_linear_oral_1_lag`: Calculate C(t) for a 3-compartment linear model after a single oral dose
- `calc_sd_3cmt_linear_infusion`: Calculate C(t) for a 3-compartment linear model after a single IV infusion
- `calc_sd_3cmt_linear_oral_0`: Calculate C(t) for a 3-compartment linear model after a single dose, with zero-order absorption

- calc_sd_3cmt_linear_oral_0_lag: Calculate C(t) for a 3-compartment linear model after a single dose, with zero-order absorption and a lag time
- calc_sd_3cmt_linear_oral_1: Calculate C(t) for a 3-compartment linear model after a single oral dose

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

References

Bertrand J & Mentre F (2008). Mathematical Expressions of the Pharmacokinetic and Pharmacodynamic Models implemented in the Monolix software. http://lixoft.com/wp-content/uploads/2016/03/PKPDlibrary.pdf

Rowland M, Tozer TN. Clinical Pharmacokinetics and Pharmacodynamics: Concepts and Applications (4th). Lippincott Williams & Wilkins, Philadelphia, 2010.

Examples

```
Ct <- calc_sd_3cmt_linear_bolus(t = 11.75, CL = 3.5, V1 = 20, V2 = 500,
    V3 = 200, Q2 = 0.5, Q3 = 0.05, dose = 100)
Ct <- calc_sd_3cmt_linear_oral_1_lag(t = 11.75, CL = 3.5, V1 = 20, V2 = 500,
    V3 = 200, Q2 = 0.5, Q3 = 0.05, ka = 1, dose = 100, tlag = 1.5)
Ct <- calc_sd_3cmt_linear_infusion(t = 11.75, CL = 3.5, V1 = 20, V2 = 500,
    V3 = 200, Q2 = 0.5, Q3 = 0.05, dose = 100, tinf=1)
Ct <- calc_sd_3cmt_linear_oral_0(t = 11.75, CL = 3.5, V1 = 20, V2 = 500,
    V3 = 200, Q2 = 0.5, Q3 = 0.05, dur = 1, dose = 100)
Ct <- calc_sd_3cmt_linear_oral_0_lag(t = 11.75, CL = 3.5, V1 = 20, V2 = 500,
    V3 = 200, Q2 = 0.5, Q3 = 0.05, dur = 1, dose = 100, tlag=1.5)
Ct <- calc_sd_3cmt_linear_oral_1(t = 11.75, CL = 3.5, V1 = 20, V2 = 500,
    V3 = 200, Q2 = 0.5, Q3 = 0.05, ka = 1, dose = 100)
```

calc_ss_1cmt *Calculate C(t) for a 1-compartment linear model at steady-state*

Description

Calculate C(t) for a 1-compartment linear model at steady-state

Usage

```
calc_ss_1cmt(tad, tau, dose, dur = NULL, tinf = NULL, ...)

calc_ss_1cmt_linear_bolus(tad, tau, dose, ...)

calc_ss_1cmt_linear_infusion(tad, tau, dose, tinf, ...)

calc_ss_1cmt_linear_oral_0(tad, tau, dose, dur, ...)
```

```
calc_ss_1cmt_linear_oral_0_lag(tad, tau, dose, dur, ...)

calc_ss_1cmt_linear_oral_1_lag(tad, tau, dose, ...)

calc_ss_1cmt_linear_oral_1(tad, tau, dose, ...)
```

Arguments

`tad`	Time after dose (h)
`tau`	Dosing interval (h)
`dose`	Dose
`dur`	Duration of zero-order absorption (h)
`tinf`	Duration of infusion (h)
`...`	Passed to ‘calc_derived_1cpt()‘

Value

Concentration of drug at requested time (`t`) after a single dose, given provided set of parameters and variables.

Functions

- `calc_ss_1cmt_linear_bolus`: Calculate C(t) for a 1-compartment linear model with IV bolus dosing at steady state
- `calc_ss_1cmt_linear_infusion`: Calculate C(t) for a 1-compartment linear model with infusion at steady state
- `calc_ss_1cmt_linear_oral_0`: Calculate C(t) for a 1-compartment linear model with zero-order oral absorption at steady state
- `calc_ss_1cmt_linear_oral_0_lag`: Calculate C(t) for a 1-compartment linear model with zero-order oral absorption at steady state, with lag time
- `calc_ss_1cmt_linear_oral_1_lag`: Calculate C(t) for a 1-compartment linear model with first-order oral absorption at steady state, with lag time
- `calc_ss_1cmt_linear_oral_1`: Calculate C(t) for a 1-compartment linear model with first-order oral absorption at steady state

Author(s)

Justin Wilkins, <`justin.wilkins@occams.com`>

References

Bertrand J & Mentre F (2008). Mathematical Expressions of the Pharmacokinetic and Pharmacodynamic Models implemented in the Monolix software. `http://lixoft.com/wp-content/uploads/2016/03/PKPDlibrary.pdf`

Rowland M, Tozer TN. Clinical Pharmacokinetics and Pharmacodynamics: Concepts and Applications (4th). Lippincott Williams & Wilkins, Philadelphia, 2010.

Examples

```
Ct <- calc_ss_1cmt_linear_bolus(t=0:24, CL=6, V=25, dose=600, tau=24)
Ct <- calc_ss_1cmt_linear_infusion(tad=0:36, CL=2, V=25, dose=600, tinf=1, tau=24)
Ct <- calc_ss_1cmt_linear_oral_0(tad=0:36, CL=2, V=25, dose=600, dur=1, tau=24)
Ct <- calc_ss_1cmt_linear_oral_0_lag(tad=0:36, CL=2, V=25, dose=600, dur=1, tau=24, tlag=1.5)
Ct <- calc_ss_1cmt_linear_oral_1_lag(tad=0:36, CL=2, V=25, dose=600,
    ka=0.25, tlag=0.75, tau=24)
Ct <- calc_ss_1cmt_linear_oral_1(tad=0:36, CL=2, V=25, dose=600, ka=0.25, tau=24)
```

calc_ss_2cmt *Calculate C(t) for a 2-compartment linear model at steady-state*

Description

Calculate C(t) for a 2-compartment linear model at steady-state

Usage

```
calc_ss_2cmt(tad, tau, dose, dur = NULL, tinf = NULL, ...)

calc_ss_2cmt_linear_bolus(tad, tau, dose, ...)

calc_ss_2cmt_linear_infusion(tad, tau, dose, tinf, ...)

calc_ss_2cmt_linear_oral_0(tad, tau, dose, dur, ...)

calc_ss_2cmt_linear_oral_1_lag(tad, tau, dose, ...)

calc_ss_2cmt_linear_oral_0_lag(tad, tau, dose, dur, ...)

calc_ss_2cmt_linear_oral_1(tad, tau, dose, ...)
```

Arguments

`tad`	Time after dose (h)
`tau`	Dosing interval (h)
`dose`	Dose
`dur`	Duration of zero-order absorption (h)
`tinf`	Duration of infusion (h)
`...`	Passed to ‘calc_derived_2cpt()‘

Value

Concentration of drug at requested time (`t`) at steady-state, given provided set of parameters and variables.

Functions

- `calc_ss_2cmt_linear_bolus`: Calculate C(t) for a 2-compartment linear model with IV bolus dosing at steady-state
- `calc_ss_2cmt_linear_infusion`: Calculate C(t) for a 2-compartment linear model with infusion at steady state
- `calc_ss_2cmt_linear_oral_0`: Calculate C(t) for a 2-compartment linear model at steady-state with zero-order oral dosing
- `calc_ss_2cmt_linear_oral_1_lag`: Calculate C(t) for a 2-compartment linear model at steady-state with first-order oral dosing
- `calc_ss_2cmt_linear_oral_0_lag`: Calculate C(t) for a 2-compartment linear model at steady-state with zero-order oral dosing and a lag time
- `calc_ss_2cmt_linear_oral_1`: Calculate C(t) for a 2-compartment linear model at steady-state with first-order oral dosing

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

References

Bertrand J & Mentre F (2008). Mathematical Expressions of the Pharmacokinetic and Pharmacodynamic Models implemented in the Monolix software. http://lixoft.com/wp-content/uploads/2016/03/PKPDlibrary.pdf

Rowland M, Tozer TN. Clinical Pharmacokinetics and Pharmacodynamics: Concepts and Applications (4th). Lippincott Williams & Wilkins, Philadelphia, 2010.

Examples

```
Ct <- calc_ss_2cmt_linear_bolus(tad = 11.75, CL = 7.5, V1 = 20, V2 = 30, Q = 0.5,
    dose = 10, tau=24)
Ct <- calc_ss_2cmt_linear_infusion(tad = 11.75, CL = 7.5, V1 = 20, V2 = 30, Q = 0.5,
    dose = 10, tinf = 1, tau = 12)
Ct <- calc_ss_2cmt_linear_oral_0(tad = 23, CL = 2.5, V1 = 20, V2 = 30, Q = 0.5,
    dose = 1000, dur = 1, tau = 24)
Ct <- calc_ss_2cmt_linear_oral_1_lag(tad = 11.75, CL = 7.5, V1 = 20, V2 = 30, Q = 0.5,
    dose = 1000, ka = 1, tau=24, tlag=2)
Ct <- calc_ss_2cmt_linear_oral_0_lag(tad = 23, CL = 2.5, V1 = 20, V2 = 30, Q = 0.5,
    dose = 1000, dur = 1, tau = 24, tlag=2)
Ct <- calc_ss_2cmt_linear_oral_1(tad = 11.75, CL = 7.5, V1 = 20, V2 = 30, Q = 0.5,
    dose = 1000, ka = 1, tau=24)
```

`calc_ss_3cmt` *Calculate C(t) for a 3-compartment linear model at steady-state*

Description

Calculate C(t) for a 3-compartment linear model at steady-state

Usage

```
calc_ss_3cmt(tad, tau, dose, dur = NULL, tinf = NULL, ...)

calc_ss_3cmt_linear_bolus(tad, tau, dose, ...)

calc_ss_3cmt_linear_oral_1_lag(tad, tau, dose, ...)

calc_ss_3cmt_linear_infusion(tad, tau, dose, tinf, ...)

calc_ss_3cmt_linear_oral_0(tad, tau, dose, dur, ...)

calc_ss_3cmt_linear_oral_0_lag(tad, tau, dose, dur, ...)

calc_ss_3cmt_linear_oral_1(tad, tau, dose, ...)
```

Arguments

`tad`	Time after dose (h)
`tau`	Dosing interval (h)
`dose`	Dose
`dur`	Duration of zero-order absorption (h)
`tinf`	Duration of infusion (h)
`...`	Passed to ‘calc_derived_3cpt()‘

Value

Concentration of drug at requested time (`t`) at steady-state, given provided set of parameters and variables.

Functions

- `calc_ss_3cmt_linear_bolus`: Calculate C(t) for a 3-compartment linear model at steady state with IV bolus dosing
- `calc_ss_3cmt_linear_oral_1_lag`: Calculate C(t) for a 3-compartment linear model at steady-state with first-order oral dosing with a lag time
- `calc_ss_3cmt_linear_infusion`: Calculate C(t) for a 3-compartment linear model at steady state with IV infusions

- `calc_ss_3cmt_linear_oral_0`: Calculate C(t) for a 3-compartment linear model at steady state, with zero-order absorption
- `calc_ss_3cmt_linear_oral_0_lag`: Calculate C(t) for a 3-compartment linear model at steady state, with zero-order absorption and lag time
- `calc_ss_3cmt_linear_oral_1`: Calculate C(t) for a 3-compartment linear model at steady-state with first-order oral dosing

Author(s)

Justin Wilkins, <`justin.wilkins@occams.com`>

References

Bertrand J & Mentre F (2008). Mathematical Expressions of the Pharmacokinetic and Pharmacodynamic Models implemented in the Monolix software. `http://lixoft.com/wp-content/uploads/2016/03/PKPDlibrary.pdf`

Rowland M, Tozer TN. Clinical Pharmacokinetics and Pharmacodynamics: Concepts and Applications (4th). Lippincott Williams & Wilkins, Philadelphia, 2010.

Examples

```
Ct <- calc_ss_3cmt_linear_bolus(t = 11.75, CL = 3.5, V1 = 20, V2 = 500,
    V3 = 200, Q2 = 0.5, Q3 = 0.05, dose = 100, tau=24)
Ctrough <- calc_ss_3cmt_linear_oral_1_lag(t = 11.75, CL = 3.5, V1 = 20, V2 = 500,
    V3 = 200, Q2 = 0.5, Q3 = 0.05, ka = 1, dose = 100, tau=24, tlag = 1.5)
Ct <- calc_ss_3cmt_linear_infusion(tad = 11.75, CL = 2.5, V1 = 20, V2 = 50,
    V3 = 100, Q2 = 0.5, Q3 = 0.05, dose = 1000, tinf=1, tau=24)
Ct <- calc_ss_3cmt_linear_oral_0(tad = 11.75, CL = 3.5, V1 = 20, V2 = 500,
    V3 = 200, Q2 = 0.5, Q3 = 0.05, dur = 1, dose = 100, tau = 24)
Ct <- calc_ss_3cmt_linear_oral_0_lag(tad = 11.75, CL = 3.5, V1 = 20, V2 = 500,
    V3 = 200, Q2 = 0.5, Q3 = 0.05, dur = 1, dose = 100, tau = 24, tlag = 1.5)
Ct <- calc_ss_3cmt_linear_oral_1(tad = 11.75, CL = 3.5, V1 = 20,
    V2 = 500, V3 = 200, Q2 = 0.5, Q3 = 0.05, ka = 1, dose = 100, tau = 24)
```

`get_auc` *Calculate the area under the curve (AUC) for each subject over the time interval for dependent variables (`dv`) using the trapezoidal rule.*

Description

Calculate the area under the curve (AUC) for each subject over the time interval for dependent variables (dv) using the trapezoidal rule.

Usage

```
get_auc(data, time = "TIME", id = "ID", dv = "DV")
```

Arguments

`data` A data frame.

`time` A string containing the name of the chronologically ordered time variable in `data`.

`id` A string containing the name of the ID column (defining subject level data) in `data`.

`dv` A string containing the name of the dependent variable column in `data`.

Value

A data frame containing one AUC value for every subject as defined by `id`.

Based on the `AUC` function originally written by Leonid Gibiansky in package MIfuns 5.1, from Metrum Institute.

Author(s)

Leonid Gibiansky, `<lgibiansky@quantpharm.com>`

References

https://code.google.com/archive/p/mifuns/

Examples

```
## Not run:
 AUCs <- get_auc(myAUCdata)

## End(Not run)
```

`get_est_table` *Create a table of model parameter estimates from a NONMEM output object.*

Description

Create a table of model parameter estimates from a NONMEM output object.

Usage

```
get_est_table(
  x,
  thetaLabels = c(),
  omegaLabels = c(),
  sigmaLabels = c(),
  sigdig = 3
)
```

Arguments

`x`	A NONMEM output object generated using `read_nm`.
`thetaLabels`	A vector containing labels for THETA parameters.
`omegaLabels`	A vector containing labels for OMEGA parameters.
`sigmaLabels`	A vector containing labels for SIGMA parameters.
`sigdig`	The desired number of significant digits to display.

Value

A named vector of NONMEM model parameter estimates.

Author(s)

Justin Wilkins, <`justin.wilkins@occams.com`>

See Also

NONMEM (`http://www.iconplc.com/innovation/nonmem/`)

Examples

```
## Not run:
 nmOutput <- read_nm("run315.xml")
 estTab   <- get_est_table(nmOutput)

## End(Not run)
```

get_omega *Extract variability parameter estimates from a NONMEM output object.*

Description

Extract variability parameter estimates from a NONMEM output object.

Usage

```
get_omega(x, output = "est", sigdig = 6, sep = "-", est.step = NULL)
```

Arguments

`x` A NONMEM output object generated using `read_nm`.

`output` A flag specifying the matrix or matrices to be output. Valid flag values are `est` (the default), `se`, `rse`, `cor`, `cse`, `95ci`, or `all`.

`sigdig` Specifies the number of significant digits to be provided (default=6).

`sep` Specifies the separator character to use for 95% confidence intervals (default="-").

`est.step` Specifies which estimation step to return parameters from (default is the last).

Value

A symmetrical matrix, or a list of symmetrical matrices if `all` is specified.

Author(s)

Justin Wilkins, `<justin.wilkins@occams.com>`

`est` returns the estimated OMEGA variance-covariance matrix. `se` returns the standard errors for the estimated OMEGA variance-covariance matrix. `rse` returns the relative standard errors for the estimated OMEGA variance-covariance matrix (se/est*100). `cor` returns the correlation matrix matrix. `cse` returns the standard errors for the correlation matrix. `95ci` returns the asymptotic 95% confidence intervals for the elements of the OMEGA variance-covariance matrix (est +/- 1.96*se). `all` returns all available OMEGA matrices.

See Also

NONMEM (`http://www.iconplc.com/innovation/nonmem/`)

Examples

```
## Not run:
 nmOutput  <- read_nm("run315.xml")
 omegas    <- get_omega(nmOutput)
 omegaRSEs <- get_omega(nmOutput, "rse")

## End(Not run)
```

`get_shrinkage` *Extract shrinkage estimates from a NONMEM output object.*

Description

Extract shrinkage estimates from a NONMEM output object.

Usage

```
get_shrinkage(x, output = "eta", type = "sd", sigdig = 3, est.step = NULL)
```

Arguments

x	A NONMEM output object generated using read_nm.
output	A flag specifying the shrinkage estimates to be output. Valid flag values are eta (the default), epsilon, or all.
type	Specifies the type of shrinkage to report. Valid values are sd (standard deviation, the default) or vr (variance, if present in the XML output).
sigdig	Specifies the number of significant digits to be provided (default=3).
est.step	Specifies which estimation step to return parameters from (default is the last).

Value

A named vector of NONMEM shrinkage estimates, or in the case of all, a list of named vectors.

eta returns a vector of ETA shrinkages, as reported by NONMEM. epsilon returns EPSILON shrinkage, as reported by NONMEM. all returns both ETA and EPSILON shrinkage estimates as a list of vectors.

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

See Also

NONMEM (http://www.iconplc.com/innovation/nonmem/)

Examples

```
## Not run:
 nmOutput <- read_nm("run315.xml")
 shr <- get_shrinkage(nmOutput, output="all")

## End(Not run)
```

get_sigma *Extract residual variability parameter estimates from a NONMEM output object.*

Description

Extract residual variability parameter estimates from a NONMEM output object.

Usage

```
get_sigma(x, output = "est", sigdig = 6, sep = "-", est.step = NULL)
```

Arguments

x A NONMEM output object generated using read_nm.

output A flag specifying the matrix or matrices to be output. Valid flag values are est (the default), se, rse, cor, cse, 95ci, or all.

sigdig Specifies the number of significant digits to be provided (default=6).

sep Specifies the separator character to use for 95% confidence intervals (default="-").

est.step Specifies which estimation step to return parameters from (default is the last).

Value

A symmetrical matrix, or a list of symmetrical matrices if all is specified.

est returns the estimated SIGMA variance-covariance matrix. se returns the standard errors for the estimated SIGMA variance-covariance matrix. rse returns the relative standard errors for the estimated SIGMA variance-covariance matrix (se/est*100). cor returns the correlation matrix matrix. cse returns the standard errors for the correlation matrix. 95ci returns the asymptotic 95% confidence intervals for the elements of the SIGMA variance-covariance matrix (est +/- 1.96*se). all returns all available SIGMA matrices.

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

See Also

NONMEM (http://www.iconplc.com/innovation/nonmem/)

Examples

```
## Not run:
 nmOutput <- read_nm("run315.xml")
 sigmas   <- get_sigma(nmOutput)
 sigmaRSEs <- get_sigma(nmOutput, "rse")

## End(Not run)
```

get_theta *Extract structural model parameter estimates and associated information from a NONMEM output object.*

Description

Extract structural model parameter estimates and associated information from a NONMEM output object.

Usage

```
get_theta(x, output = "est", sigdig = 6, sep = "-", est.step = NULL)
```

Arguments

x	A NONMEM output object generated using read_nm.
output	A flag specifying the matrix or matrices to be output. Valid flag values are est (the default), se, rse, 95ci, or all.
sigdig	Specifies the number of significant digits to be provided (default=6).
sep	Specifies the separator character to use for 95% confidence intervals (default="-").
est.step	Specifies which estimation step to return parameters from (default is the last).

Value

A named vector of NONMEM model parameter estimates, or in the case of all, a list of named vectors.

est returns a vector of THETA values. se returns a vector of THETA standard errors. rse returns a vector of THETA relative standard errors (se/est*100). 95ci returns a vector of the asymptotic 95% confidence intervals for the elements of THETA (est +/- 1.96*se). all returns all available THETA information as a list of named vectors.

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

See Also

NONMEM (http://www.iconplc.com/innovation/nonmem/)

Examples

```
## Not run:
 nmOutput <- read_nm("run315.xml")
 thetas <- get_theta(nmOutput)

## End(Not run)
```

`gm` *Calculate geometric mean*

Description

Calculate geometric mean

Usage

```
gm(x)
```

Arguments

`x`	Numeric vector.

Value

The geometric mean. `NA` is returned if there are any non-positive elements in `x`.

Author(s)

Justin Wilkins, `<justin.wilkins@occams.com>`

Examples

```
gm(c(0.5, 7, 8, 5))
```

`pcv` *Calculate percentage coefficient of variation*

Description

Calculate percentage coefficient of variation

Usage

```
pcv(x, na.rm = FALSE)
```

Arguments

`x`	Numeric vector.
`na.rm`	A logical value indicating whether NA values should be stripped before the computation proceeds.

Value

The percentage coefficient of variation.

Author(s)

Justin Wilkins, <`justin.wilkins@occams.com`>

Examples

```
pcv(rnorm(50, 5, 7.56))
```

`pk_curve` *Provide concentration-time curves.*

Description

Provide concentration-time curves.

Usage

```
pk_curve(
  t,
  model = "1cmt_oral",
  params = list(ka = 2.77, CL = 2.5, V = 25),
  dose = 600,
  ii = 24,
  addl = 0,
  ss = F
)
```

Arguments

`t` Observation time in h, specified as a vector.

`model` The model to use. Must be one of "1cmt_bolus", "1cmt_infusion", "1cmt_oral", "2cmt_bolus", "2cmt_infusion", "2cmt_oral", "3cmt_bolus", "3cmt_infusion", "3cmt_oral". The default is "1cmt_oral".

`params` A named list containing parameter values for the selected model type.

`dose` Dose amount.

`ii` Interdose interval (or tau), in hours (default 24).

`addl` Number of additional doses (default 0).

`ss` Assume steady state concentration (default `FALSE`).

Value

A data frame containing times (`t`) and concentrations (`cp`).

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

Examples

```
plot(pk_curve(t=seq(0,72,by=0.1), model="3cmt_oral", ii=12, addl=5,
  params=list(CL=2.5, V1=25, V2=2, V3=5, Q2=0.5, Q3=0.25, ka=1)), type="l")
```

plot_nmprogress *Plot NONMEM parameter estimation by iteration.*

Description

plot_nmprogress returns a plot or set of plots showing the evolution of parameter estimates by iteration.

Usage

```
plot_nmprogress(
  fileName,
  fileExt = ".lst",
  metric = "perc",
  lineCol = "#902C10",
  idlineCol = "black"
)
```

Arguments

`fileName` A NONMEM output file prefix, without extension (e.g. 'run315').

`fileExt` The file extension for NONMEM output, set to '.lst' by default.

`metric` What to show in the plot. Allowed options are 'est' (the actual estimate) or 'perc' (the percentage change in the estimated or OFV since estimation began). Default is 'perc'.

`lineCol` Line color. Default is '#902C10'.

`idlineCol` Identity line color (only used if 'perc' metric is selected). Default is black.

Value

A set of plots.

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

See Also

NONMEM (http://www.iconplc.com/innovation/nonmem/)

Examples

```
## Not run:
plot_nmprogress("run315")
plot_nmprogress("run315", ".nmlst")

## End(Not run)
```

plot_scm *Plot PsN SCM results.*

Description

`plot_scm` returns a step-plot summary of a Perl-speaks-NONMEM (PsN, https://uupharmacometrics.github.io/PsN/) SCM (stepwise covariate modeling) procedure. It depends on the presence of an `scmlog.txt` file in the specified directory, and is inspired by the plot code provided with PsN 4.8.1.

Usage

```
plot_scm(
  dir,
  phase = "both",
  lineCol = "#902C10",
  lineType = 1,
  lineSize = 1,
  pointCol = "#902C10",
  pointShape = 16,
  pointSize = 3,
  textCol = "black",
  textSize = 5
)
```

Arguments

`dir`	A PsN SCM folder (containing `scmlog.txt`).
`phase`	SCM phase. Can be `"both"` (the default), `"forward"` or `"backward"`.
`lineCol`	Line color. Default is ’#902C10’.
`lineType`	Line type. Default is ’1’.
`lineSize`	Line color. Default is ’1’.
`pointCol`	Point color. Default is ’#902C10’.
`pointShape`	Point shape. Default is ’16’.
`pointSize`	Point size. Default is ’3’.
`textCol`	Point color. Default is ’black’.
`textSize`	Point color. Default is ’6’.

Value

A ggplot2 plot object.

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

See Also

NONMEM (http://www.iconplc.com/innovation/nonmem/)

Lindbom L, Ribbing J & Jonsson EN (2004). Perl-speaks-NONMEM (PsN) - A Perl module for NONMEM related programming. Computer Methods and Programs in Biomedicine, 75(2), 85-94. https://doi.org/10.1016/j.cmpb.2003.11.003

Lindbom L, Pihlgren P & Jonsson N (2005). PsN-Toolkit - A collection of computer intensive statistical methods for non-linear mixed effect modeling using NONMEM. Computer Methods and Programs in Biomedicine, 79(3), 241-257. https://doi.org/10.1016/j.cmpb.2005.04.005

Examples

```
## Not run:
scm <- plot_scm("E:/DrugX/ModelDevelopment/scm310")

## End(Not run)
```

---

read_nm   *Read NONMEM 7.2+ output into a list of lists.*

Description

Read NONMEM 7.2+ output into a list of lists.

Usage

```
read_nm(fileName)
```

Arguments

fileName	A NONMEM XML output file (e.g. "run315.xml").

Value

A list of lists corresponding to a NONMEM output object.

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

See Also

NONMEM (http://www.iconplc.com/innovation/nonmem/)

Examples

```
## Not run:
nmOutput <- read_nm("run315.xml")

## End(Not run)
```

read_nmcov *Read in the NONMEM variance-covariance matrix.*

Description

Read in the NONMEM variance-covariance matrix.

Usage

```
read_nmcov(fileName)
```

Arguments

`fileName`	Root filename for the NONMEM run (e.g. "run315"). This function reads the ".cov" NONMEM output table, and will return an error if this is missing.

Value

A symmetrical variance-covariance matrix covering all model parameters.

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

See Also

NONMEM (http://www.iconplc.com/innovation/nonmem/)

Examples

```
## Not run:
nmVcov <- read_nmcov("run315")

## End(Not run)
```

read_nmext *Read NONMEM output into a list.*

Description

read_nmext returns a summary of a given NONMEM run, including termination messages, parameter estimates, and precision estimates. Minimally, the NONMEM output and '.ext' files must be available.

Usage

```
read_nmext(fileName, fileExt = ".lst")
```

Arguments

fileName	A NONMEM output file prefix, without extension (e.g. "run315").
fileExt	The file extension for NONMEM output, set to ".lst" by default.

Value

A list of lists, containing 'Termination' (summary of NONMEM's termination output, including shrinkages and ETABAR estimates), 'OFV' (the objective function value), 'Thetas' (a vector of structural parameter estimates, or THETAs), 'Omega', a list of lists containing the OMEGA matrix, 'Sigma', a list of lists containing the SIGMA matrix, 'seThetas', a vector of standard errors for THETAs, 'seOmega', a list of lists containing standard errors for the OMEGA matrix, and 'seSigma', a list of lists containing standard errors for the SIGMA matrix.

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

See Also

NONMEM (http://www.iconplc.com/innovation/nonmem/)

Examples

```
## Not run:
read_nmext("run315")
read_nmext("run315", ".nmlst")

## End(Not run)
```

`read_nmtables` *Reads NONMEM output tables.*

Description

Reads NONMEM output tables.

Usage

```
read_nmtables(
  tableFiles = NULL,
  runNo = NULL,
  tabSuffix = "",
 tableNames = c("sdtab", "mutab", "patab", "catab", "cotab", "mytab", "extra", "xptab"),
  quiet = FALSE,
  ...
)
```

Arguments

`tableFiles`	NONMEM table files to be read.
`runNo`	Run number.
`tabSuffix`	Table file suffix.
`tableNames`	List of root table names, using the Xpose naming convention as the default.
`quiet`	Flag for displaying intermediate output (defaults to `FALSE`).
`...`	Additional arguments.

Value

A data frame.

Note

Adapted from Xpose 4 (`https://CRAN.R-project.org/package=xpose4`).

Author(s)

Justin Wilkins, Niclas Jonsson, Andrew Hooker

See Also

NONMEM (`http://www.iconplc.com/innovation/nonmem/`)

Jonsson EN, Karlsson MO. Xpose–an S-PLUS based population pharmacokinetic/pharmacodynamic model building aid for NONMEM. Comput Methods Programs Biomed. 1999 Jan;58(1):51-64

Examples

```
## Not run:
tables <- read_nmtables(315)

## End(Not run)
```

read_scm *Read PsN SCM output into a format suitable for further use.*

Description

`read_scm` returns a summary of a Perl-speaks-NONMEM (PsN, https://uupharmacometrics.github.io/PsN/) SCM (stepwise covariate modeling) procedure. It depends on the presence of an `scmlog.txt` file in the specified directory.

Usage

```
read_scm(dir, startPhase = "forward")
```

Arguments

`dir` A PsN SCM folder (containing `scmlog.txt`).

`startPhase` Where to start collating the output; can be `"forward"` (the default) or `"backward"`.

Value

A list of data frames, containing

`forward` all models evaluated during the forward inclusion step of covariate model building

`forwardSummary` the covariate relationships selected at each forward step

`forwardP` the P-value used for inclusion during the forward inclusion step

`backward` all models evaluated during the backward elimination step of covariate model building

`backwardSummary` the covariate relationships eliminated at each backward step

`backwardP` the P-value used for exclusion during the backward elimination step

Author(s)

Justin Wilkins, `<justin.wilkins@occams.com>`

See Also

NONMEM (http://www.iconplc.com/innovation/nonmem/)

Lindbom L, Ribbing J & Jonsson EN (2004). Perl-speaks-NONMEM (PsN) - A Perl module for NONMEM related programming. Computer Methods and Programs in Biomedicine, 75(2), 85-94. https://doi.org/10.1016/j.cmpb.2003.11.003

Lindbom L, Pihlgren P & Jonsson N (2005). PsN-Toolkit - A collection of computer intensive statistical methods for non-linear mixed effect modeling using NONMEM. Computer Methods and Programs in Biomedicine, 79(3), 241-257. https://doi.org/10.1016/j.cmpb.2005.04.005

Examples

```
## Not run:
scm <- read_scm("E:/DrugX/ModelDevelopment/scm310")

## End(Not run)
```

`rnm` *Read NONMEM 7.2+ output into an R object.*

Description

Read NONMEM 7.2+ output into an R object.

Usage

```
rnm(
  index,
  prefix = "run",
  pathNM,
  ndig = 3,
  ndigB = 3,
  ndigP = 1,
  Pci = 95,
  ext = ".lst",
  extmod = ".mod",
  Pvalues = TRUE,
  RawCI = FALSE,
  ...
)
```

Arguments

`index` The NONMEM model index, i.e. the numeric part of the filename assuming it follows the convention 'run123.mod'.

`prefix` The NONMEM model prefix, assuming it follows the convention 'run123.mod'. The default is `"run"`.

`pathNM` The path to the NONMEM output. This should not contain a trailing slash.

`ndig` Number of significant digits to use. The default is 3.

`ndigB` Number of significant digits to use. The default is 3.

`ndigP` Number of digits after the decimal point to use for percentages. The default is 1.

`Pci` Asymptotic confidence interval to apply when reporting parameter uncertainty. The default is 95.

`ext` NONMEM output file extension. The default is `".lst"`.

`extmod` NONMEM control stream file extension. The default is `"mod"`.

`Pvalues` Report P-values for parameters? The default is `TRUE`.

`RawCI` Report confidence intervals without estimate? The default is `FALSE`.

`...` Additional arguments.

Details

The output list is composed of the following objects:

- "Theta"A data frame describing the structural (fixed-effect) parameters, containing parameter name, estimated value, standard error (SE), coefficient of variation (CV), lower and upper confidence limits (CIL and CIU, based on `Pci`), and P-value, calculated as `2 * (1 -pnorm(abs(theta/theta.se)))`.
- "Eta"A data frame describing the interindividual random-effects parameters, containing estimated value, standard error (SE), coefficient of variation (CV, calculated as `abs(100*(SE/OMEGA))`), coefficient of variation (EtaCV, calculated as `100*sqrt(OMEGA)`), and shrinkage.
- "Epsilon"A data frame describing the residual random-effects parameters, containing estimated value, standard error (SE), coefficient of variation (CV, calculated as `abs(100*(SE/OMEGA))`), coefficient of variation (EtaCV, calculated as `100*sqrt(SIGMA)`), and shrinkage.
- "CorTheta"A data frame containing the correlation matrix for fixed effects (`"THETA"`).
- "CorOmega"A data frame containing the correlation matrix for interindividual random effects (`"OMEGA"`).
- "CorSigma"A data frame containing the correlation matrix for residual random effects (`"OMEGA"`).
- "OmegaMatrix"A data frame containing the `"OMEGA"` matrix.
- "SigmaMatrix"A data frame containing the `"OMEGA"` matrix.
- "CovMatrixTheta"A data frame containing the variance-covariance matrix for structural parameters (`THETA`).
- "CovMatrix"A data frame containing the complete variance-covariance matrix.
- "OFV"The objective function value.
- "ThetaString"A data frame containing all relevant fixed-effects parameter information, suitable for use in a table of parameter estimates. Contains parameter name, estimate, standard error, coefficient of variation, combined estimate and asymptotic confidence interval, and P-value.

- "EtaString"A data frame containing all relevant interindivudal random-effects parameter information, suitable for use in a table of parameter estimates. Contains parameter name, estimate (variance), standard error, coefficient of variation, percentage value (calculated as `100*sqrt(OMEGA)`), and shrinkage.
- "EpsString"A data frame containing all relevant residual random-effects parameter information, suitable for use in a table of parameter estimates. Contains parameter name, estimate (variance), standard error, coefficient of variation, percentage value (calculated as `100*sqrt(SIGMA)`), and shrinkage.
- "RunTime"Run time.
- "ConditionN"Condition number.

Value

A list containing information extracted from the NONMEM output.

Author(s)

Rik Schoemaker, <rik.schoemaker@occams.com>

Justin Wilkins, <justin.wilkins@occams.com>

See Also

NONMEM (http://www.iconplc.com/innovation/nonmem/)

Examples

```
## Not run:
nmOutput <- rnm("run315.lst")

## End(Not run)
```

sample_omega — *Sample from the multivariate normal distribution using the OMEGA variance-covariance matrix to generate new sets of simulated ETAs from NONMEM output.*

Description

Sample from the multivariate normal distribution using the OMEGA variance-covariance matrix to generate new sets of simulated ETAs from NONMEM output.

Usage

```
sample_omega(nmRun, n, seed)
```

Arguments

nmRun	Root filename for the NONMEM run (e.g. "run315").
n	Number of samples required.
seed	Random seed.

Value

A data frame containing n samples from the multivariate normal distribution, using the estimated NONMEM OMEGA variance-covariance matrix. This provides n sets of ETA estimates suitable for simulation of new patients.

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

See Also

NONMEM (http://www.iconplc.com/innovation/nonmem/)

Examples

```
## Not run:
 omDist <- sample_omega("run315", 5000, seed=740727)

## End(Not run)
```

sample_sigma

Sample from the multivariate normal distribution using the SIGMA variance-covariance matrix to generate new sets of simulated EPSILONs from NONMEM output.

Description

Sample from the multivariate normal distribution using the SIGMA variance-covariance matrix to generate new sets of simulated EPSILONs from NONMEM output.

Usage

```
sample_sigma(nmRun, n, seed)
```

Arguments

nmRun	Root filename for the NONMEM run (e.g. "run315").
n	Number of samples required.
seed	Random seed.

Value

A data frame containing n samples from the multivariate normal distribution, using the estimated NONMEM SIGMA variance-covariance matrix. This provides n sets of EPSILON estimates suitable for simulation of new datasets.

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

See Also

NONMEM (http://www.iconplc.com/innovation/nonmem/)

Examples

```
## Not run:
 sigDist <- sample_sigma("run315", 5000, seed=740727)

## End(Not run)
```

sample_uncert *Sample from the multivariate normal distribution to generate new sets of parameters from NONMEM output.*

Description

Sample from the multivariate normal distribution to generate new sets of parameters from NONMEM output.

Usage

```
sample_uncert(nmRun, n, seed)
```

Arguments

nmRun	Root filename for the NONMEM run (e.g. "run315").
n	Number of samples required.
seed	Random seed.

Value

A data frame containing n samples from the multivariate normal distribution, using NONMEM typical parameter estimates the NONMEM variance-covariance matrix (from the *.cov file). This provides n sets of parameter estimates sampled from the uncertainty distribution, suitable for simulation under model uncertainty.

Author(s)

Justin Wilkins, <justin.wilkins@occams.com>

See Also

NONMEM (http://www.iconplc.com/innovation/nonmem/)

Examples

```
## Not run:
 nmMatrix <- sample_uncert("run315", 5000, seed=740727)

## End(Not run)
```

table_rtf *Read NONMEM output into a list.*

Description

table_rtf generates an RTF table from a data frame.

Usage

```
table_rtf(
  df,
  outFile = NULL,
  rtfFile = TRUE,
  boldHeader = TRUE,
  rowNames = FALSE,
  ...
)
```

Arguments

df A data frame.

outFile A filename for writing the table to. If NULL, writes to console.

rtfFile If TRUE (the default), then add RTF tabs to generate a fully formatted RTF file.

boldHeader If TRUE, make the header bold.

rowNames If TRUE, include row names in the table. Default is FALSE.

... Other formatting options for the table body.

Value

An RTF table based on the data frame provided.

Author(s)

John Johnson, <johndjohnson@gmail.com>

References

http://www.r-bloggers.com/another-solution-to-the-r-to-word-table-problem/

Examples

```
## Not run:
scm <- read_scm("E:/DrugX/ModelDevelopment/scm310")
myRTF <- table_rtf(scm$forwardSummary)

## End(Not run)
```

Index