Package ‘pmxTools’

February 21, 2023

Type Package

Title Pharmacometric and Pharmacokinetic Toolkit

Version 1.3

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

RoxygenNote 7.2.3

Imports ggplot2, chron, xml2, dplyr (>= 0.8.5), tibble, gghalves, ggdist, scales, MASS, stringr, PKNCA, magrittr, data.tree

Depends R (>= 3.50), patchwork

Suggests testthat, vdiffr, knitr, rmarkdown

Encoding UTF-8

VignetteBuilder knitr

Config/testthat/edition 3

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].
Satyaprakash Nayak [ctb],
Leonid Gibiansky [ctb],
Andrew Hooker [ctb],
E. Niclas Jonsson [ctb],
Mats O. Karlsson [ctb],
John Johnson [ctb]

Repository CRAN

Date/Publication 2023-02-21 16:00:08 UTC

R topics documented:

blq_trans ......................................................... 3
breaks_blq_general .............................................. 4
calc_derived ..................................................... 5
calc_sd_1cmt ..................................................... 8
calc_sd_2cmt ..................................................... 10
calc_sd_3cmt ..................................................... 11
calc_ss_1cmt ..................................................... 13
calc_ss_2cmt ..................................................... 15
calc_ss_3cmt ..................................................... 17
count_na ......................................................... 18
dgr_table ......................................................... 19
estimate_lloq ................................................... 20
fmt_signif ....................................................... 21
ftrans_blq_linear .............................................. 22
gcv ............................................................... 22
gcv_convert ..................................................... 23
get_auc ........................................................... 24
get_est_table ..................................................... 25
get_omega ........................................................ 26
get_probinfo ...................................................... 27
get_shrinkage .................................................... 28
get_sigma ........................................................ 29
get_theta ........................................................ 30
gm ................................................................. 31
itrans_blq_linear ............................................... 32
label_blq ........................................................ 32
pcv ............................................................... 33
pk_curve ........................................................ 34
plot_dist ......................................................... 35
plot_nmprogress ............................................... 38
plot_scm ........................................................ 39
read_nm .......................................................... 42
read_nmcov ...................................................... 43
read_nmext ...................................................... 44
read_nmtables .................................................. 45
read_nm_all ..................................................... 46
blq_trans

A transform for ggplot2 with data that may be below the lower limit of quantification

Description

If the lloq is not provided, it will be estimated from the data as the minimum value above zero.

Usage

```r
blq_trans(lloq, x, multiplier = 0.5, lloq_text)
```

```r
blq_log_trans(lloq, x, multiplier = 0.5, base = 10, lloq_text)
```

Arguments

- `lloq`: The value of the lower limit of quantification as a numeric scalar
- `x`: (only used if `lloq` is missing), the data for `lloq` estimation.
- `multiplier`: When data are < `lloq`, they are replaced by `lloq*multiplier` for display.
- `lloq_text`: The text to use on the axis to indicate values < `lloq`. It will be automatically set to `paste0("<", lloq)` if missing.
- `base`: The base for the logarithm

Value

A "trans" object based on the scales package for BLQ data.

Functions

- `blq_log_trans()`: Log-scale transformation with BLQ

See Also

Other BLQ Transformation: `breaks_blq_general()`, `estimate_lloq()`, `ftrans_blq_linear()`, `itrans_blq_linear()`, `label_blq()`
Examples

```r
## Not run:
library(ggplot2)
ggplot(data=data.frame(x=1:10, y=1:10), aes(x=x, y=y)) + geom_point()

ggplot(data=data.frame(x=1:10, y=1:10), aes(x=x, y=y)) + geom_point() + scale_x_continuous(trans=blq_trans(lloq=3))

## End(Not run)
```

breaks_blq_general  Generate breaks for measurements below the limit of quantification

Description

Breaks that are < lloq are removed. If the lowest break is removed if it is too close to the lloq.

Usage

```r
breaks_blq_general(lloq, breakfun, trans = identity, ...)
```

Arguments

- `lloq`: The value of the lower limit of quantification as a numeric scalar
- `breakfun`: The function used for normal scale breaks if the lloq were not present.
- `trans`: A parameter translation function (typically either identity for linear scale or log for log scale).
- `...`: passed as breakfun(n=n, ...)

Details

For ggplot2 scales. This is not usually used directly. See blq_trans() and blq_log10_trans() for the functions that are more commonly used.

Value

A function for calculating breaks with arguments `x` and `n`

See Also

Other BLQ Transformation: blq_trans(), estimate_lloq(), ftrans_blq_linear(), itrans_blq_linear(), label_blq()
Examples

breaks_blq_general(lloq=3, breakfun=scales::breaks_extended)(1:100, n=5)

calcDerived

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

calcDerived(..., verbose = FALSE)

calcDerived_1cpt(
  CL,
  V = NULL,
  V1 = NULL,
  ka = NULL,
  dur = NULL,
  tlag = NULL,
  tinf = NULL,
  dose = NULL,
  tau = NULL,
  step = 0.1,
  type = "all",
  sigdig = 5
)

calcDerived_2cpt(
  CL,
  V1 = NULL,
  V2,
  Q2 = NULL,
  V = NULL,
  Q = NULL,
  dur = NULL,
  tinf = NULL,
  ka = NULL,
  tlag = NULL,
  dose = NULL,
  tau = NULL,
  step = 0.1,
  type = "all",
  sigdig = 5
)
calc_derived_3cpt(
    CL,
    V1 = NULL,
    V2,
    V3,
    Q2 = NULL,
    Q3,
    V = NULL,
    Q = NULL,
    ka = NULL,
    dur = NULL,
    tinf = NULL,
    tlag = NULL,
    dose = NULL,
    tau = NULL,
    step = 0.1,
    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 (volume per time units, e.g. L/h)
V1, V Central volume of distribution (volume units, e.g. L). Values are synonyms; use only one.
ka Absorption rate (inverse time units, e.g. 1/h)
dur Duration of zero-order absorption (time units, e.g. h)
tlag Absorption lag time (time units, e.g. h)
tinf Duration of infusion (time units, e.g. h)
dose Dose (amount units, e.g. mg)
tau Duration of interdose interval (time units, e.g. h; defaults to 24)
step Time increment to use when estimating NCA parameters (time units, e.g. h; defaults to 0.1)
type Parameters to return. Default is "all". If not "all", this may be a vector from the names of the return value list.
sigdig Number of significant digits to be returned. Default is 5.
V2 First peripheral volume of distribution (volume units, e.g. L)
Q2, Q Intercompartmental clearance from central to first peripheral compartment (volume per time units, e.g. L/h). Values are synonyms; use only one.
V3 Second peripheral volume of distribution (volume units, e.g. L)
Q3 Intercompartmental clearance from central to second peripheral compartment (volume per time units, e.g. 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. If a dose is provided, estimated non-compartmental analysis (NCA) parameters will be provided as well, based on simulation of single-dose and (if ‘tau’ is specified) steady-state time courses.

- **Vss**: Volume of distribution at steady state, $V_{ss}$ (volume units, e.g. L); 1-, 2-, and 3-compartment
- **$k_{10}$**: First-order elimination rate, $k_{10}$ (inverse time units, e.g. 1/h); 1-, 2-, and 3-compartment
- **$k_{12}$**: First-order rate of transfer from central to first peripheral compartment, $k_{12}$ (inverse time units, e.g. 1/h); 2- and 3-compartment
- **$k_{21}$**: First-order rate of transfer from first peripheral to central compartment, $k_{21}$ (inverse time units, e.g. 1/h); 2- and 3-compartment
- **$k_{13}$**: First-order rate of transfer from central to second peripheral compartment, $k_{13}$ (inverse time units, e.g. 1/h); 3-compartment
- **$k_{31}$**: First-order rate of transfer from second peripheral to central compartment, $k_{31}$ (inverse time units, e.g. 1/h); 3-compartment
- **$t_{1/2,\alpha}$**: $t_{1/2,\alpha}$ (time units, e.g. h); 1-, 2-, and 3-compartment
- **$t_{1/2,\beta}$**: $t_{1/2,\beta}$ (time units, e.g. h); 2- and 3-compartment
- **$t_{1/2,\gamma}$**: $t_{1/2,\gamma}$ (time units, e.g. h); 3-compartment
- **$\alpha$**: $\alpha$; 1-, 2-, and 3-compartment
- **$\beta$**: $\beta$; 2- and 3-compartment
- **$\gamma$**: $\gamma$; 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
- **AUCinf**: Area under the concentration-time curve to infinity (single dose)
- **AUCtau**: Area under the concentration-time curve over the dosing interval at steady state
- **Cmax**: Maximum concentration after a single dose
- **Cmaxss**: Maximum concentration over the dosing interval at steady state
- **Tmax**: Time after dose of maximum concentration
- **AUCinf_dose_normalized**: Dose-normalized area under the concentration-time curve to infinity (single dose)
- **AUCtau_dose_normalized**: Dose-normalized area under the concentration-time curve over the dosing interval at steady state
- **Cmax_dose_normalized**: Dose-normalized maximum concentration after a single dose
- **Cmaxss_dose_normalized**: Dose-normalized maximum concentration over the dosing interval at steady state
- **step**: Time increment used when estimating NCA parameters.

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

**Author(s)**

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

Bill Denney, <wdenney@humanpredictions.com>

**References**

Shafer S. L. CONVERT.XLS


**Examples**

```r
params <- calc_derived(CL=29.4, V1=23.4, V2=114, V3=4614, Q2=270, Q3=73)
params <- calc_derived_1cmt(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**

```r
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, ...)
```
calc_sd_1cmt

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>
Bill Denney, <wdenney@humanpredictions.com>

References


Examples

```r
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 2-compartment linear model*

**Description**

Calculate C(t) for a 2-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_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_3cmt

- 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_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>
Bill Denney, <wdenney@humanpredictions.com>

References


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, tlag = 1)
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 3-compartment linear model**

Description

Calculate C(t) for a 3-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 'calcDerived_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>
Bill Denney, <wdenney@humanpredictions.com>
References


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>
Bill Denney, <wdenney@humanpredictions.com>

References


calc_ss_2cmt

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)

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
 tau
dose
dur
 tinf
 ...  Time after dose (h)
 Dosing interval (h)
 Dose
 Duration of zero-order absorption (h)
 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


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)
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


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)

count_na

Count the number of NA values in a vector.

Description

Count the number of NA values in a vector.

Usage

count_na(x)

Arguments

x  A vector.
**dgr_table**

Generate a summary table of descriptive data for every individual in a dataset suitable for tabulation in a report.

### Description

Generate a summary table of descriptive data for every individual in a dataset suitable for tabulation in a report.

### Usage

```r
dgr_table(
  dat,
  fields,
  names,
  cutoff = 7,
  sig = 3,
  by = NULL,
  idvar = "ID",
  navars = c("-99", "-999")
)
```

### Arguments

- **dat**: An input data frame, with one row per unique individual.
- **fields**: A vector of strings containing the names of the fields to be included in the summary table.
- **names**: A vector of strings containing descriptive names for the fields to be included in the summary table.

### Examples

```r
## Not run:
  count_na(c(0,5,7,NA,3,3,NA))
## End(Not run)
```
cutoff  An integer defining the maximum number of unique values a variable should have to be considered categorical. Fields with more than this number of unique values are considered continuous for the purposes of the summary table (defaults to 7).

sig  The number of significant digits summary values should have (defaults to 3).

by  The field to use for grouping (a string). If not NULL (the default), the summary table will contain columns for each unique value of this field, as well as a column summarizing across all fields.

idvar  The field in the dataset identifying each unique individual (defaults to "ID").

navars  A vector containing values that are to be interpreted as missing (defaults to ".-99" and ".-999"). ‘NA’ values are always considered to be missing.

Value

A data frame containing a summary of all the fields listed in fields, for each individual in the dataset (the dataset should not contain duplicated individuals), conditioned on the field in by. Continuous values are summarized as median, mean, range and number of missing values. Categorical values are summarized as count and relative percentage.

Author(s)

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

Examples

```r
## Not run:
count_na(c(0,5,7,NA,3,3,NA))

## End(Not run)
```

---

**estimate_lloq**  
*Estimate the lower limit of quantification (LLOQ) from a vector*

Description

Nonnegative values are considered to be above the LLOQ. NA values are ignored.

Usage

```r
estimate_lloq(x)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>The numeric vector to use for estimation of the LLOQ</td>
</tr>
</tbody>
</table>
Value

The lowest, nonzero value from x. If all are NA or zero, 1 is returned, and a warning is issued.

See Also

Other BLQ Transformation: `blq_trans()`, `breaks_blq_general()`, `ftrans_blq_linear()`, `itrans_blq_linear()`, `label_blq()`

Examples

```r
estimate_lloq(c(NA, 0, 2, 5))
```

### fmt_signif

Format a number with the correct number of significant digits and trailing zeroes.

#### Usage

```r
fmt_signif(x, digits = 3)
```

#### Arguments

- **x**: A vector of numeric values.
- **digits**: The number of significant digits values should have (defaults to 3).

#### Value

A string containing the properly-formatted number.

#### Author(s)

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

#### Examples

```r
## Not run:
fmt_signif(c(36.44, 0.0002, 3336.7), digits=3)
## End(Not run)
```
ftrans_blq_linear  

Forward transformation for linear BLQ data

Description

For ggplot2 scales.

Usage

ftrans_blq_linear(lloq, multiplier)

ftrans_blq_log(lloq, multiplier, base = 10)

Arguments

lloq  
The value of the lower limit of quantification as a numeric scalar

multiplier  
When data are $<$ lloq, they are replaced by lloq*multiplier for display.

base  
The base for the logarithm

Value

A function of x that replaces $x < lloq$ with lloq*multiplier

Functions

- ftrans_blq_log(): Log-scale transformation

See Also

Other BLQ Transformation: blq_trans(), breaks_blq_general(), estimate_lloq(), itrans_blq_linear(), label_blq()

gcv  

Calculate a geometric coefficient of variation.

Description

Calculate a geometric coefficient of variation.

Usage

gcv(x, na.rm = F, neg.rm = F)
Arguments

- **x**
  A vector.
- **na.rm**
  Flag for removing NA values (defaults to FALSE).
- **neg.rm**
  Flag for removing negative or zero values (defaults to FALSE).

Value

The geometric coefficient of variation of the input vector. If `neg.rm` is FALSE and values <= 0 are present, NA will be returned.

Author(s)

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

Examples

```r
## Not run:
gcv(myvector)
## End(Not run)
```

---

**gcv_convert**

Convert geometric variance or standard deviation to a geometric coefficient of variation

Description

The equation used is: 100*sqrt(exp(gvar)-1)

Usage

```r
gcv_convert(gvar = gsd^2, gsd)
```

Arguments

- **gvar**
  The geometric variance (note that this is the variance not a vector of values to compute the gcv from)
- **gsd**
  The geometric standard deviation

Value

Geometric coefficient of variation

Author(s)

Bill Denney
get_auc

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/
get_est_table

Examples

```r
## 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

```r
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 (https://www.iconplc.com/innovation/nonmem/)
### Description

Extract variability parameter estimates from a NONMEM output object.

### Usage

```r
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. `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 (https://www.iconplc.com/innovation/nonmem/)
## get_probinfo

Extract problem and estimation information from a NONMEM output object.

### Description

Extract problem and estimation information from a NONMEM output object.

### Usage

```r
get_probinfo(x, sigdig = 6, est.step = NULL)
```

### Arguments

- `x`: A NONMEM output object generated using `read_nm`.
- `sigdig`: Specifies the number of significant digits to be provided (default=6).
- `est.step`: Specifies which estimation step to return parameters from (default is the last).

### See Also


### Examples

```r
## Not run:
nmOutput <- read_nm("run315.xml")
probInfo <- get_probinfo(nmOutput)
## End(Not run)
```
get_shrinkage

Extract shrinkage estimates from a NONMEM output object.

Description

Extract shrinkage estimates from a NONMEM output object.

Usage

```r
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 (https://www.iconplc.com/innovation/nonmem/)

Examples

```r
## 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.
- `cor` returns the correlation 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.
- `all` returns all available SIGMA matrices.

Author(s)

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

See Also

NONMEM (https://www.iconplc.com/innovation/nonmem/)
Examples

```r
## 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

```r
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>
### Description

Calculate geometric mean

### Usage

```r
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

```r
gm(c(0.5, 7, 8, 5))
```
**itrans_blq_linear**  
*Inverse transformation for linear BLQ data*

**Description**
For ggplot2 scales.

**Usage**

\[
\text{itrans_blq_linear}(lloq)
\]

\[
\text{itrans_blq_log}(lloq, \text{base})
\]

**Arguments**

- `lloq`  
The value of the lower limit of quantification as a numeric scalar  
- `base`  
The base for the logarithm

**Value**
A function of \(x\) that replaces \(x < lloq\) with \(lloq\)

**Functions**
- `itrans_blq_log()`: Log-scale inverse transform

**See Also**
Other BLQ Transformation:  
- `blq_trans()`, `breaks_blq_general()`, `estimate_lloq()`, `ftrans_blq_linear()`, `label_blq()`

---

**label_blq**  
*Label axes with censoring labels for BLQ*

**Description**
For ggplot2 scales.

**Usage**

\[
\text{label_blq}(lloq, lloq\_text)
\]

**Arguments**

- `lloq`  
The value of the lower limit of quantification as a numeric scalar  
- `lloq\_text`  
The text to use on the axis to indicate values \(< lloq\). It will be automatically set to `paste0("<", lloq)` if missing. 
`pcv`  

Value  
A function of x which returns the formatted values.

See Also  
Other BLQ Transformation: `blq_trans()`, `breaks_blq_general()`, `estimate_lloq()`, `ftrans_blq_linear()`, `itrans_blq_linear()`

---

**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_dist

Plot a distribution as a hybrid containing a halfeye, a boxplot and jittered points.

Description

Plot a distribution as a hybrid containing a halfeye, a boxplot and jittered points.

Usage

plot_dist(
  dat,
  yvar,
  xvar = NULL,
  ylim = NULL,
  xlb = "",
  ylb = "",
  identity_line = FALSE,
  identity_value = 0,
  he_adjust = 0.5,
  he_width = 0.6,
  he_justification = -0.2,
  he_col = "black",
  he_fill = "#F8766D",
  he_alpha = 0.9,
  he_slab_type = "pdf",
  he_breaks = "Sturges",
  he_outline_bars = FALSE,
  he_point_interval = "median_qi",
  bxp_width = 0.12,
  bxp_outlier_col = NA,
  bxp_outlier_fill = NA,
  bxp_outlier_shape = 19,
  bxp_outlier_size = 1.5,
  bxp_col = "black",
  bxp_fill = "#F8766D",
  bxp_alpha = 0.9,
  bxp_notch = FALSE,
  bxp_notchwidth = 0.5,
  hp_range_scale = 0.4,
  hp_alpha = 0.25,
  hp_col = "#F8766D",
  hp_transformation = position_jitter(),
  na.rm = FALSE
)
Arguments

dat  A data frame.
yvar  The name of the field containing values to be plotted.
xvar  The name of the field containing the grouping variable (defaults to ‘NULL’).
ylim  Limits for the y-axis. Defaults to NULL. If provided, should be a 2-element vector containing the upper and lower limits.
xlb  Label for the x-axis.
ylb  Label for the y-axis.
identity_line  Show a line of identity? Default FALSE.
identity_value  If an identity line is shown, it will be drawn horizontally at this y-value (default 0).
he_adjust  If he_slab_type is "pdf", bandwidth for the density estimator is adjusted by multiplying it by this value.
he_width  Width of the halfeye component of the plot (default 0.6).
he_justification  Justification of the halfeye component of the plot (default -0.2).
he_col  Color for the halfeye component of the plot.
he_fill  Fill color for the halfeye component of the plot.
he_alpha  Alpha for the halfeye component of the plot (default 0.9).
he_slab_type  The type of slab function to calculate for the halfeye component of the plot: probability density (or mass) function ("pdf", the default), cumulative distribution function ("cdf"), complementary CDF ("ccdf") or histogram ("histogram").
he_breaks  If slab_type is "histogram", the breaks parameter that is passed to hist() to determine where to put breaks in the histogram.
he_outline_bars  If slab_type is "histogram", determines if outlines in between the bars are drawn when the slab_color aesthetic is used. If FALSE (the default), the outline is drawn only along the tops of the bars; if TRUE, outlines in between bars are also drawn.
he_point_interval  A function from the ggdist::point_interval family (e.g., median_qi, mean_qi, mode_hdi, etc), or a string giving the name of a function from that family (e.g., "median_qi", "mean_qi", "mode_hdi", etc. This function determines the point summary (typically mean, median, or mode) and interval type (quantile interval, qi; highest-density interval, hdi; or highest-density continuous interval, hdci). Output will be converted to the appropriate x- or y-based aesthetics depending on the value of orientation.
bxp_width  Width of the boxplot component (default 0.12).
bxp_outlier_col  Color for outliers in the boxplot component.
bxp_outlier_fill  Fill color for outliers in the boxplot component.
plot_dist

bxp_outlier_shape
Shape for outliers in the boxplot component.

bxp_outlier_size
Size for outliers in the boxplot component.

bxp_col
Color for the boxplot component.

bxp_fill
Fill color for the boxplot component.

bxp_alpha
Alpha for the boxplot component.

bxp_notch
If FALSE (default) make a standard box plot. If TRUE, make a notched box plot. Notches are used to compare groups; if the notches of two boxes do not overlap, this suggests that the medians are significantly different.

bxp_notchwidth
For a notched box plot, width of the notch relative to the body (default 0.5).

hp_range_scale
If no 'width' argument is specified in hp_transformation, used to determine the width of the jitter. Defaults to 0.75, which is half of the allotted space for the jitter-points, whereas 1 would use all of the allotted space.

hp_alpha
Alpha for the jitter.

hp_col
Color for the jitter.

hp_transformation
An evaluated position_*() function yielding a ‘Position’ object with specified parameters to calculate the transformation of the points. Defaults to ggplot2::position_jitter.

na.rm
If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.

Value

A plot containing jittered points, a boxplot and a density plot or histogram illustrating the distribution of every group of the data under evaluation.

Author(s)

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

Examples

## Not run:
plot_dist(dat, "ETA1", identity_line = T, he_slab_type = "histogram", he_breaks = 30)

## End(Not run)
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 (https://www.iconplc.com/innovation/nonmem/)

Examples

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

## End(Not run)
```
**plot_scm**

Visualize PsN SCM output.

**Description**

`plot_scm` returns a visualization of a Perl-speaks-NONMEM (PsN, [https://uupharmacometrics.github.io/PsN/](https://uupharmacometrics.github.io/PsN/)) SCM (stepwise covariate modeling) procedure. It depends on the presence of `scmlog.txt` and `short_scmlog.txt` files in the specified directory.

**Usage**

```r
plot_scm(
  dir,
  startPhase = "forward",
  fwdSuccessCol = "#66C2A5",
  fwdFailCol = "black",
  bwdSuccessCol = "#FC8D62",
  bwdFailCol = "black",
  defCol = "black",
  fwdSuccessFillCol = "#B3E2CD",
  fwdFailFillCol = "white",
  bwdSuccessFillCol = "#FDCDAC",
  bwdFailFillCol = "white",
  defFillCol = "white",
  fwdSuccessFontCol = "black",
  fwdFailFontCol = "black",
  bwdSuccessFontCol = "black",
  bwdFailFontCol = "black",
  defFontCol = "black",
  fullFwdCol = "#8DA0CB",
  finalCol = "#E78AC3",
  fullFwdFillCol = "#CBD5E8",
  finalFillCol = "#F4CAE4",
  fullFwdFontCol = "black",
  finalFontCol = "black",
  fullFwdWidth = "2px",
  finalWidth = "2px",
  defWidth = "1px",
  nodeStyle = "filled,rounded",
  nodeShape = "box",
  fontname = "helvetica",
  rankdir = "TB",
  layout = "dot",
  lookupDF = NULL,
  ...
)
```
Arguments

dir  A PsN SCM folder (containing scmlog.txt and short_scmlog.txt).
startPhase  Where to start collating the output; can be "forward" (the default) or "backward".
fwdSuccessCol  Node outline color for a model fit matching the forward inclusion criterion.
fwdFailCol  Node outline color for a model fit not matching the forward inclusion criterion.
bwdSuccessCol  Node outline color for a model fit matching the backward elimination criterion.
bwdFailCol  Node outline color for a model fit not matching the backward elimination criterion.
defCol  Default node outline color.
fwdSuccessFillCol  Node fill color for a model fit matching the forward inclusion criterion.
fwdFailFillCol  Node fill color for a model fit not matching the forward inclusion criterion.
bwdSuccessFillCol  Node fill color for a model fit matching the backward elimination criterion.
bwdFailFillCol  Node fill color for a model fit not matching the backward elimination criterion.
defFillCol  Default node fill color.
fwdSuccessFontCol  Node font color for a model fit matching the forward inclusion criterion.
fwdFailFontCol  Node font color for a model fit not matching the forward inclusion criterion.
bwdSuccessFontCol  Node font color for a model fit matching the backward elimination criterion.
bwdFailFontCol  Node font color for a model fit not matching the backward elimination criterion.
defFontCol  Default node font color.
fullFwdCol  Node outline color for the full forward model (i.e. the final model before the backward elimination procedure in SCM).
finalCol  Node outline color for the final reduced model (i.e. the final model reached after the backward elimination procedure in SCM).
fullFwdFillCol  Node fill color for the full forward model (i.e. the final model before the backward elimination procedure in SCM).
finalFillCol  Node fill color for the final reduced model (i.e. the final model reached after the backward elimination procedure in SCM).
fullFwdFontCol  Node font color for the full forward model (i.e. the final model before the backward elimination procedure in SCM).
finalFontCol  Node font color for the final reduced model (i.e. the final model reached after the backward elimination procedure in SCM).
fullFwdWidth  Node outline width for the full forward model (i.e. the final model before the backward elimination procedure in SCM).
finalWidth  Node outline width for the final reduced model (i.e. the final model reached after the backward elimination procedure in SCM).
defWidth  Default node outline width.
nodeStyle  Node style. A string containing a comma-separated list of options (which include "filled", "striped", "wedged", "diagonals" and "rounded"). See the GraphViz documentation for further details.
	nodeShape  Node shape. Options include "box" (the default), "oval", "diamond", "egg", "plaintext", "point", "square", "triangle" and many more. See the GraphViz documentation for further details.

tfontname  Font for nodes. Options depend heavily on the local system - see the GraphViz documentation for further details.
	rankdir  Direction of graph layout. Possible values are "TB" (the default), "LR", "BT", "RL", corresponding to directed graphs drawn from top to bottom, from left to right, from bottom to top, and from right to left, respectively.

layout  Graph layout. Possible values are "dot" (the default), "neato", "twopi", and "circo". Note that of these, "dot" is the easiest to interpret and the others may produce odd results.

lookupDF  A data frame containing a lookup table for node labels. By default, plot_scm will use the PSN model names. If a lookup table containing the fields 'Model' and 'Alias' is provided, model names in 'Model' will be replaced in the output plots by matching labels in 'Alias'.

...  Additional parameters passed to the underlying SetNodeStyle and SetEdgeStyle functions, which in turn rely on DiagrammeR.

Details

This function parses PsN SCM output and displays it as a GraphViz graph (effectively, an HTML widget). It is built on plot.Node - please refer to documentation for this function for a more detailed overview of what is possible (a lot). For more specific details, see http://rich-iannone.github.io/DiagrammeR/docs.html.

Value

A grViz object.

Author(s)

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

See Also

NONMEM (https://www.iconplc.com/innovation/nonmem/)
GraphViz (https://graphviz.org/Documentation.php)
Other NONMEM reading: read_nm_all(), read_nm_multi_table(), read_nmcov(), read_nmext(), read_nmtables(), read_nm(), read_scm()
## read_nm

### Description

Read NONMEM 7.2+ output into a list of lists.

### Usage

```r
read_nm(fileName, directory = NULL, quiet = FALSE, ...)
```

### Arguments

- **fileName**
  - A NONMEM XML output file (e.g. "run315.xml").
- **directory**
  - The directory to look for files within. If NULL, uses the current directory.
- **quiet**
  - Flag for displaying intermediate output.
- **...**
  - Passed to each of the read functions (ignored in the functions).

### Value

A list of lists corresponding to a NONMEM output object.

### Author(s)

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

### See Also


Other NONMEM reading: `plot_scm()`, `read_nm_all()`, `read_nm_multi_table()`, `read_nmcov()`, `read_nmext()`, `read_nmtables()`, `read_scm()`

### Examples

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

## End(Not run)
```
Description

Read in the NONMEM variance-covariance matrix.

Usage

read_nmcov(fileName, quiet = FALSE, directory = NULL, ...)

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.
quiet     Flag for displaying intermediate output.
directory The directory to look for files within. If NULL, uses the current directory.
...       Passed to each of the read functions (ignored in the functions).

Value

A symmetrical variance-covariance matrix covering all model parameters.

Author(s)

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

See Also

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

Other NONMEM reading: plot_scm(), read_nm_all(), read_nm_multi_table(), read_nmext(),
read_nmtables(), read_nm(), read_scm()

Examples

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

## End(Not run)
```
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",
    directory = NULL,
    quiet = FALSE,
    estNo = NULL,
    ...
)

Arguments

fileName A NONMEM output file prefix, without extension (e.g. "run315").
fileExt The file extension for NONMEM output, set to ".lst" by default.
directory The directory to look for files within. If NULL, uses the current directory.
quiet Flag for displaying intermediate output.
estNo The estimation number to report (by default, if only one estimation step is present, that will be reported; if multiple are reported, the last will be reported by default).
...
Passed to each of the read functions (ignored in the functions).

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


Other NONMEM reading: `plot_scm()`, `read_nm_all()`, `read_nm_multi_table()`, `read_nmcov()`, `read_nmtables()`. `read_nm()`, `read_scm()`

Examples

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

## End(Not run)
```

---

### read_nmtables

*Reads NONMEM output tables.*

**Description**

Reads NONMEM output tables.

**Usage**

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

**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.
- `directory`: The directory to look for files within. If NULL, uses the current directory.
- `output_type`: Should output be a "data.frame" where all results are merged or a "list" of data.frames.
- `...`: Passed to each of the read functions (ignored in the functions).
Value

A data.frame or list of data.frames depending on the output_type argument.

Note

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

Author(s)

Bill Denney, Justin Wilkins, Niclas Jonsson, Andrew Hooker

References

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

See Also

Other NONMEM reading: plot_scm(), read_nm_all(), read_nm_multi_table(), read_nmcov(), read_nnext(), read_nm(), read_scm()

Examples

## Not run:
    tables <- read_nmtables(runNo=315)

## End(Not run)

read_nm_all

Read all NONMEM files for a single NONMEM run.

Description

Read all NONMEM files for a single NONMEM run.

Usage

read_nm_all(runNo, run_prefix = "run", directory = NULL, quiet = FALSE, ...)

Arguments

runNo

Run number.

run_prefix

The start to the name of the run.

directory

The directory to look for files within. If NULL, uses the current directory.

quiet

Flag for displaying intermediate output.

... 

Passed to each of the read functions (ignored in the functions).
The filename for loading is constructed as `paste(run_prefix, runNo)`. To load a nonstandard file, simply set one of those values to `NULL`.

See Also

Other NONMEM reading: `plot_scm()`, `read_nm_multi_table()`, `read_nmcov()`, `read_nmext()`, `read_nmtables()`, `read_nm()`, `read_scm()`
**Examples**

```r
## Not run:
read_nm_multi_table("run1.cov", row.names=1)

## End(Not run)
```

---

**read_nm_std_ext**  
*Read a standard NONMEM extension file*

**Description**

Read a standard NONMEM extension file

**Usage**

```r
read_nm_std_ext(fileName, extension, directory = NULL, ...)
```

**Arguments**

- `fileName`: The filename (with directory name, if applicable) to read (with or without the extension)
- `extension`: The file extension to optionally append (preferably starting with a ".")
- `directory`: The directory to look for files within. If NULL, uses the current directory.
- `...`: Passed to `read_nm_multi_table()`

**Value**

NULL if the file does not exist or the value of `read_nm_multi_table()` if it does exist.

**Examples**

```r
## Not run:
read_nm_std_ext("run1", "phi")

## 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 scmlog.txt and short_scmlog.txt files in the specified directory.

**Usage**

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

**Arguments**

- `dir` A PsN SCM folder (containing scmlog.txt and short_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 (https://www.iconplc.com/innovation/nonmem/)


Other NONMEM reading: `plot_scm()`, `read_nm_all()`, `read_nm_multi_table()`, `read_nmcov()`, `read_nmext()`, `read_nmtables()`, `read_nm()`
Examples

```r
## 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**

```r
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 $2 \times (1 - \text{pnorm(abs(theta/theta.se)))}$.

• "Eta" A data frame describing the interindividual random-effects parameters, containing estimated value, standard error (SE), coefficient of variation (CV, calculated as $100 \times (SE/OMEGA)$), coefficient of variation (EtaCV, calculated as $100 \times \sqrt{OMEGA}$), and shrinkage.

• "Epsilon" A data frame describing the residual random-effects parameters, containing estimated value, standard error (SE), coefficient of variation (CV, calculated as $100 \times (SE/OMEGA)$), coefficient of variation (EtaCV, calculated as $100 \times \sqrt{SIGMA}$), 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 interindividual 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 $100 \times \sqrt{OMEGA}$), 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 $100 \times \sqrt{SIGMA}$), 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 (https://www.iconplc.com/innovation/nonmem/)

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

```r
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.
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**

```r
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**

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.xml").
- **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 (https://www.iconplc.com/innovation/nonmem/)

## Examples
```r
## Not run:
nmMatrix <- sample_uncert("run315.xml", 5000, seed=740727)
## End(Not run)
```
table_rtf

Description

table_rtf generates an RTF table from a data frame.

Usage

    table_rtf(
        df,              # A data frame.
        outFile = NULL, # A filename for writing the table to. If NULL, writes to console.
        rtfFile = TRUE,  # If TRUE (the default), then add RTF tabs to generate a fully formatted RTF file.
        boldHeader = TRUE,  # If TRUE, make the header bold.
        rowNames = FALSE,  # If TRUE, include row names in the table. Default is FALSE.
        ...               # Other formatting options for the table body.
    )

Arguments

    df
    outFile
    rtfFile
    boldHeader
    rowNames
    ...

Value

An RTF table based on the data frame provided.

Author(s)

John Johnson, <johndjohnson@gmail.com>

References


Examples

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

    ## End(Not run)
Index

* BLQ Transformation
  blq_trans, 3
  breaks_blq_general, 4
  estimate_lloq, 20
  ftrans_blq_linear, 22
  itrans_blq_linear, 32
  label_blq, 32
  blq_log_trans (blq_trans), 3
  breaks_blq_general, 3, 4, 21, 22, 32, 33
  calc_derived, 5
  calc_derived_1cpt (calc_derived), 5
  calc_derived_2cpt (calc_derived), 5
  calc_derived_3cpt (calc_derived), 5
  calc_sd_1cmt, 8
  calc_sd_1cmt_linear_bolus (calc_sd_1cmt), 8
  calc_sd_1cmt_linear_infusion (calc_sd_1cmt), 8
  calc_sd_1cmt_linear_oral_0 (calc_sd_1cmt), 8
  calc_sd_1cmt_linear_oral_0_lag (calc_sd_1cmt), 8
  calc_sd_1cmt_linear_oral_1 (calc_sd_1cmt), 8
  calc_sd_1cmt_linear_oral_1_lag (calc_sd_1cmt), 8
  calc_sd_2cmt, 10
  calc_sd_2cmt_linear_bolus (calc_sd_2cmt), 10
  calc_sd_2cmt_linear_infusion (calc_sd_2cmt), 10
  calc_sd_2cmt_linear_oral_0 (calc_sd_2cmt), 10
  calc_sd_2cmt_linear_oral_0_lag (calc_sd_2cmt), 10
  calc_sd_2cmt_linear_oral_1 (calc_sd_2cmt), 10
  calc_sd_2cmt_linear_oral_1_lag (calc_sd_2cmt), 10
  calc_sd_3cmt, 11
  calc_sd_3cmt_linear_bolus (calc_sd_3cmt), 11
  calc_sd_3cmt_linear_infusion (calc_sd_3cmt), 11
  calc_sd_3cmt_linear_oral_0 (calc_sd_3cmt), 11
  calc_sd_3cmt_linear_oral_0_lag (calc_sd_3cmt), 11
  calc_sd_3cmt_linear_oral_1 (calc_sd_3cmt), 11
  calc_sd_3cmt_linear_oral_1_lag (calc_sd_3cmt), 11
  calc_ss_1cmt, 13
  calc_ss_1cmt_linear_bolus (calc_ss_1cmt), 13
  calc_ss_1cmt_linear_infusion (calc_ss_1cmt), 13
  calc_ss_1cmt_linear_oral_0 (calc_ss_1cmt), 13
  calc_ss_1cmt_linear_oral_0_lag (calc_ss_1cmt), 13
  calc_ss_1cmt_linear_oral_1 (calc_ss_1cmt), 13
  calc_ss_1cmt_linear_oral_1_lag (calc_ss_1cmt), 13
  calc_ss_2cmt, 15

* NONMEM reading
  plot_scm, 39
  read_nm, 42
  read_nm_all, 46
  read_nm_multi_table, 47
  read_nmcov, 43
  read_nmext, 44
  read_nmtables, 45
  read_scm, 49

blq_log_trans (blq_trans), 3
blq_trans, 3, 4, 21, 22, 32, 33
breaks_blq_general, 3, 4, 21, 22, 32, 33
index

calc_ss_2cmt_linear_bolus  
(calc_ss_2cmt), 15

calc_ss_2cmt_linear_infusion  
(calc_ss_2cmt), 15

calc_ss_2cmt_linear_oral_0  
(calc_ss_2cmt), 15

calc_ss_2cmt_linear_oral_0_lag  
(calc_ss_2cmt), 15

calc_ss_2cmt_linear_oral_1  
(calc_ss_2cmt), 15

calc_ss_2cmt_linear_oral_1_lag  
(calc_ss_2cmt), 15

calc_ss_3cmt, 17

calc_ss_3cmt_linear_bolus  
(calc_ss_3cmt), 17

calc_ss_3cmt_linear_infusion  
(calc_ss_3cmt), 17

calc_ss_3cmt_linear_oral_0  
(calc_ss_3cmt), 17

calc_ss_3cmt_linear_oral_0_lag  
(calc_ss_3cmt), 17

calc_ss_3cmt_linear_oral_1  
(calc_ss_3cmt), 17

calc_ss_3cmt_linear_oral_1_lag  
(calc_ss_3cmt), 17

count_na, 18

dgr_table, 19

DiagrammeR, 41

estimate_lloq, 3, 4, 20, 22, 32, 33

fmt_signif, 21

ftrans_blq_linear, 3, 4, 21, 22, 32, 33

ftrans_blq_log (ftrans_blq_linear), 22

gcv, 22

gcv_convert, 23

gtcu, 24

get_est_table, 25

get_omega, 26

get_probinfo, 27

get_shrinkage, 28

gcmm, 29

gm, 30

get_theta, 30

ggdist::point_interval, 36

ggplot2::position_jitter, 37

itrans_blq_log (itrans_blq_linear), 32

label_blq, 3, 4, 21, 22, 32, 33

pcv, 33

pk_curve, 34

plot.Node, 41

plot_dist, 35

plot_nmprogress, 38

plot_scm, 39, 42, 43, 45–47, 49

read_nm, 25–30, 41, 42, 43, 45–47, 49

read_nm_all, 41–43, 45, 46, 47, 49

read_nm_multi_table, 41–43, 45–47, 47, 49

read_nm_std_ext, 48

read_nmcov, 41, 42, 43, 45–47, 49

read_nmext, 41–43, 44, 46, 47, 49

read_nmtables, 41–43, 45, 45, 47, 49

read_scm, 41–43, 45–47, 49

rm, 50

sample_omega, 52

sample_sigma, 53

sample_uncert, 54

SetEdgeStyle, 41

SetNodeStyle, 41

table_rtf, 55