Package ‘gastempt’

October 13, 2022

Version 0.5.5
Type Package
Title Analyzing Gastric Emptying from MRI or Scintigraphy
Description Fits gastric emptying time series from MRI or 'scintigraphic' measurements using nonlinear mixed-model population fits with 'nlme' and Bayesian methods with Stan; computes derived parameters such as t50 and AUC.
License GPL (>= 3)
NeedsCompilation yes
URL https://github.com/dmenne/gastempt
BugReports https://github.com/dmenne/gastempt/issues
Depends R (>= 4.1.0)
Imports nlme, Rcpp (>= 1.0.3), dplyr, methods, tibble (>= 3.0.0), ggplot2 (>= 3.2.0), rstan (>= 2.21.0), assertthat, stringr, shiny, utf8
Suggests rmarkdown, knitr, covr, testthat (>= 2.99), ragg, vdiffr, rstantools
LinkingTo StanHeaders (>= 2.21.0), rstan (>= 2.21.0), BH (>= 1.72.0-1), Rcpp (>= 1.0.3), RcppEigen (>= 0.3.3.7.0), RcppParallel (>= 5.0.1)
VignetteBuilder knitr
RoxygenNote 7.1.2
Encoding UTF-8
Biarch FALSE
SystemRequirements GNU make
Config/testthat/parallel true
Config/testthat/edition 3
Author Dieter Menne [aut, cre]
Maintainer Dieter Menne <dieter.menne@menne-biomed.de>
Repository CRAN
Date/Publication 2022-05-02 08:20:04 UTC
**R topics documented:**

- `coef.nlme_gastempt` .......................................................................................... 2
- `coef.stan_gastempt` ............................................................................................ 3
- `gastemptfunc` ........................................................................................................ 3
- `nlme_gastempt` ........................................................................................................ 5
- `plot.nlme_gastempt` ................................................................................................. 6
- `plot.stan_gastempt` ................................................................................................. 7
- `run_shiny` ................................................................................................................ 7
- `simulate_gastempt` .................................................................................................. 8
- `stan_gastempt` .......................................................................................................... 9
- `stan_model_names` ................................................................................................... 11
- `t50` .......................................................................................................................... 11

---

**Index** 13

---

**coef.nlme_gastempt**  
*Extract coefficients from nlme_gastempt result*

**Description**

Extract coefficients from `nlme_gastempt` result

**Usage**

```r
## S3 method for class 'nlme_gastempt'
coef(object, ...)
```

**Arguments**

- `object` Result of a call to `nlme_gastempt`
- `...` other arguments

**Value**

a data frame with coefficients. See `nlme_gastempt` for an example.
coef.stan_gastempt

Extract coefficients from stan_gastempt result

Description
Extract coefficients from stan_gastempt result

Usage
## S3 method for class 'stan_gastempt'
coef(object, ...)

Arguments

- object: Result of a call to stan_gastempt
- ...: other arguments

Value
a data frame with coefficients. See nlme_gastempt for an example.

gastemptfunc
Functions for gastric emptying analysis

Description
The linexp and the power exponential (powexp) functions can be used to fit gastric emptying curves.

Usage
linexp(t, v0 = 1, tempt = NULL, kappa = NULL, pars = NULL)
linexp_slope(t, v0 = 1, tempt = NULL, kappa = NULL, pars = NULL)
linexp_auc(v0 = 1, tempt = NULL, kappa = NULL, pars = NULL)
powexp(t, v0 = 1, tempt = NULL, beta = NULL, pars = NULL)
powexp_slope(t, v0 = 1, tempt = NULL, beta = NULL, pars = NULL)
linexp_log(t, v0 = 1, logtempt = NULL, logkappa = NULL, pars = NULL)
powexp_log(t, v0 = 1, logtempt = NULL, logbeta = NULL, pars = NULL)
Arguments

- **t**  
  Time after meal or start of scan, in minutes; can be a vector.
- **v0**  
  Initial volume at t=0.
- **tempt**  
  Emptying time constant in minutes (scalar).
- **kappa**  
  Overshoot term for linexp function (scalar).
- **pars**  
  Default NULL. If not NULL, the other parameters with exception of **t** are not used and are retrieved as named parameters from the numeric vector **pars** instead.
- **beta**  
  Power term for power exponential function (scalar).
- **logtempt**  
  Logarithm of emptying time constant in minutes (scalar).
- **logkappa**  
  Logarithm of overshoot term for linexp function (scalar).
- **logbeta**  
  Logarithm of power term for power exponential function (scalar).

Details

The `linexp` function can have an initial overshoot to model secretion.

\[
vol(t) = v_0 \times (1 + \text{kappa} \times t / \text{tempt}) \times \exp(-t / \text{tempt})
\]

The `powexp` function introduced by Elashof et al. is monotonously decreasing but has more freedom to model details in the function tail.

\[
vol(t) = v_0 \times \exp(-(t / \text{tempt}) ^ \text{beta})
\]

The `_slope` functions return the first derivatives of `linexp` and `powexp`. Use the `_log` functions to enforce positive parameters `tempt` and `beta`. Rarely required for gastric emptying curves.

Value

Vector of length(t) for computed volume.

Examples

```r
# Define parameters
kappa = 1.3
tempt = 60
v0 = 400
beta = 3
pars = c(v0 = v0, tempt = tempt, kappa = kappa)

# Plotting
oldpar = par(mfrow = c(1,3))
plot(t, linexp(t, v0, tempt, kappa), type = "l", ylab = "volume",
     main = "linexp\nkappa = 1.3 and 1.0")
lines(t, linexp(t, v0, tempt, 1), type = "l", col = "green")

# This should give the same plot as above
plot(t, linexp(t, pars = pars), type = "l", ylab = "volume",
     main = "linexp\nkappa = 1.3 and 1.0\nwith vectored parameters")
lines(t, linexp(t, v0, tempt, 1), type = "l", col = "green")

plot(t, powexp(t, v0, tempt, beta), type = "l", ylab = "volume",
     main = "powexp\nbeta = 2 and 1")
lines(t, powexp(t, v0, tempt, 1), type = "l", col = "green")
par(oldpar)
```
Simplified population fit of gastric emptying data

Description

Compute coefficients v0, tempt and kappa of a mixed model fit to a linexp function with one grouping variable.

Usage

nlme_gastempt(d, pnlsTol = 0.001, model = linexp, variant = 1)

Arguments

d A data frame with columns
   • record Record descriptor as grouping variable, e.g. patient ID
   • minute Time after meal or start of recording.
   • vol Volume of meal or stomach

pnlsTol The value of pnlsTol at the initial iteration. See nlmeControl When the model does not converge, pnlsTol is multiplied by 5 and the iteration repeated until convergence or pnlsTol >= 0.5. The effective value of pnlsTol is returned in a separate list item. When it is known that a data set converges badly, it is recommended to set the initial pnlsTol to a higher value, but below 0.5, for faster convergence.

model linexp (default) or powexp

variant For both models, there are 3 variants
   • variant = 1 The most generic version with independent estimates of all three parameters per record (random = v0 + tempt + kappa ~ 1 | record). The most likely to fail for degenerate cases. If this variant converges, use it.
   • variant = 2 Diagonal random effects (random = pdDiag(v0 + tempt + kappa) ~ 1; groups = ~record). Better convergence in critical cases. Note: I never found out why I have to use the groups parameter instead of the |; see also p. 380 of Pinheiro/Bates.
   • variant = 3 Since parameters kappa and beta respectively are the most difficult to estimate, these are fixed in this variant (random = v0 + tempt ~ 1). This variant converges in all reasonable cases, but the estimates of kappa and beta cannot be use for secondary between-group analysis. If you are only interested in t50, you can use this safe version.

Value

A list of class nlme_gastempt with elements coef, summary, plot, pnlsTol, message

   • coef is a data frame with columns:
- record Record descriptor, e.g. patient ID
- v0 Initial volume at t=0
- tempt Emptying time constant
- kappa Parameter kappa for model = linexp
- beta Parameter beta for model = powexp
- t50 Half-time of emptying
- slope_t50 Slope in t50; typically in units of ml/minute

On error, coef is NULL

- nlme_result Result of the nlme fit; can be used for addition processing, e.g. to plot residuals or via summary to extract AIC. On error, nlme_result is NULL.
- plot A ggplot graph of data and prediction. Plot of raw data is returned even when convergence was not achieved.
- pnlsTol Effective value of pnlsTo after convergence or failure.
- message String "Ok" on success, and the error message of nlme on failure.

Examples

suppressWarnings(RNGversion("3.5.0"))
set.seed(4711)
d = simulate_gastempt(n_record = 10, kappa_mean = 0.9, kappa_std = 0.3, model = linexp)$data
fit_d = nlme_gastempt(d)
# fit_d$coef # direct access
coef(fit_d) # better use accessor function
coef(fit_d, signif = 3) # Can also set number of digits
# Avoid ugly ggplot shading (not really needed...)
library(ggplot2)
theme_set(theme_bw() + theme(panel.spacing = grid::unit(0,"lines")))
# fit_d$plot # direct access is possible
plot(fit_d) # better use accessor function

plot.nlme_gastempt  Plot data points and fit curve of an nlme_gastempt fit

Description

Plot data points and fit curve of an nlme_gastempt fit

Usage

## S3 method for class 'nlme_gastempt'
plot(x, ...)

Arguments

x Result of a call to nlme_gastempt
...
other arguments
plot.stan_gastempt

**Value**

A ggplot object. Use `print()` if used non-interactively to show the curve.

---

**Description**

Plot data points and fit curve of an `stan_gastempt` fit.

**Usage**

```r
## S3 method for class 'stan_gastempt'
plot(x, ...)
```

**Arguments**

- `x`: Result of a call to `stan_gastempt`
- `...`: Other arguments

**Value**

A ggplot object. Use `print()` if used non-interactively to show the curve.

---

**Description**

Run shiny app demonstrating fit strategies with simulated data.

**Usage**

```r
run_shiny()
```

**Value**

Not used, starts shiny app.
**simulate_gastempt**  
*Simulate gastric emptying data following a linexp or powexp function*

**Description**

Simulate gastric emptying data following a linexp or powexp function

**Usage**

```r
simulate_gastempt(
  n_records = 10,
  v0_mean = 400,
  v0_std = 50,
  tempt_mean = ifelse(identical(model, linexp), 60, 120),
  tempt_std = tempt_mean/3,
  kappa_mean = 0.7,
  kappa_std = kappa_mean/3,
  beta_mean = 0.7,
  beta_std = beta_mean/3,
  noise = 20,
  student_t_df = NULL,
  missing = 0,
  model = linexp,
  seed = NULL,
  max_minute = NULL
)
```

**Arguments**

- **n_records**  
  Number of records

- **v0_mean, v0_std**  
  Mean and between record standard deviation of initial volume, typically in ml.

- **tempt_mean, tempt_std**  
  Mean and between record standard deviation of parameter $t_{empt}$, typically in minutes.

- **kappa_mean, kappa_std**  
  For linexp only: Mean and between-record standard deviation of overshoot parameter kappa. For values of kappa above 1, curve has an overshoot that can be used to follow volume time series with secretion.

- **beta_mean, beta_std**  
  For powexp only: Mean and between-record standard deviation of the so called lag parameter.

- **noise**  
  Standard deviation of normal noise when student_t_df = NULL; scaling of noise when student_t_df >= 2.
student_t_df  When NULL (default), Gaussian noise is added; when >= 2, Student_t distributed noise is added, which generates more realistic outliers. Values from 2 to 5 are useful, when higher values are used the result comes close to that of Gaussian noise. Values below 2 are rounded to 2.

missing  When 0 (default), all curves have the same number of data points. When > 0, this is the fraction of points that were removed randomly to simulate missing points. Maximum value is 0.5.

model  linexp(default) or powexp

seed  optional seed; not set if seed = NULL (default)

max_minute  Maximal time in minutes; if NULL, a sensible default rounded to hours is used

Value
A list with 3 elements:

record  Data frame with columns record(chr), v0, tempt, kappa/beta giving the effective linexp or powexp parameters for the individual record. v0 is rounded to nearest integer.

data  Data frame with columns record(chr), minute(dbl), vol(dbl) giving the time series and grouping parameters. vol is rounded to nearest integer.

stan_data  A list for use as data in Stan-based fits with elements prior_v0, n, n_record, record, minute, volume.

A comment is attached to the return value that can be used as a title

Examples

```r
suppressWarnings(RNGversion("3.5.0"))
set.seed(4711)
library(ggplot2)
vol_linexp = simulate_gastempt(n_records = 4, noise = 20)
ggplot(vol_linexp$data, aes(x = minute, y = vol)) + geom_point() + facet_wrap(~record) + ggtitle("linexp, noise = 0, no missing")

vol_powexp = simulate_gastempt(n_records = 4, missing = 0.2, student_t_df = 2)
ggplot(vol_powexp$data, aes(x = minute, y = vol)) + geom_point() + facet_wrap(~record) + ggtitle("powexp, noise = 10 (default), 20% missing, Student-t (df = 2) noise")
```

stan_gastempt  Fit gastric emptying curves with Stan

Description
Fit gastric emptying curves with Stan
stan_gastempt

Usage

stan_gastempt(
  d,
  model_name = "linexp_gastro_2b",
  lkj = 2,
  student_df = 5L,
  init_r = 0.2,
  chains = 1,
  iter = 2000,
  ...
)

Arguments

d    A data frame with columns
  • rec Record descriptor as grouping variable, e.g. patient ID
  • minute Time after meal or start of recording.
  • vol Volume of meal or stomach
model_name    Name of predefined model in gastempt/exec. Use stan_model_names() to get a list of available models.
lkj    LKJ prior for kappa/tempt correlation, only required for model linexp_gastro_2b. Values from 1.5 (strong correlation) to 50 (almost independent) are useful.
student_df    Student-t degrees of freedom for residual error; default 5. Use 3 for strong outliers; values above 10 are close to gaussian residual distribution.
init_r    for stan, default = 0.2; Stan's own default is 2, which often results in stuck chains.
chains    for stan; default = 1
iter    A positive integer specifying the number of iterations for each chain (including warmup). The default is 2000.
...
  Additional parameter passed to sampling and stan

Value

A list of class stan_gastempt with elements coef, fit, plot
  • coef is a data frame with columns:
    – rec Record descriptor, e.g. patient ID
    – v0 Initial volume at t=0
    – tempt Emptying time constant
    – kappa Parameter kappa for model = linexp
    – beta Parameter beta for model = powexp
    – t50 Half-time of emptying
    – slope_t50 Slope in t50; typically in units of ml/minute On error, coef is NULL
  • fit Result of class 'stanfit'
  • plot A ggplot graph of data and prediction. Plot of raw data is returned even when convergence was not achieved.
Examples

```r
# Runs 30+ seconds on CRAN
dd = simulate_gastempt(n_records = 6, seed = 471)
d = dd$data
ret = stan_gastempt(d)
print(ret$coef)
```

---

**stan_model_names**

*Names and descriptions of precompiled Stan models*

**Description**

By default, line 2 and 3 of comments starting with `#` or `//` in Stan file are returned

**Usage**

```r
stan_model_names(n_lines = 2, skip = 1, sep = "\n")
```

**Arguments**

- `n_lines` Number of comment lines to retrieve
- `skip` Number of lines to skip from beginning of Stan Model file
- `sep` separator for multiline strings

**Value**

A data frame with `model_name` and the first `n_lines` comment lines in model as description

---

**t50**

*Compute half-emptying time from nlme parameters*

**Description**

No closed solution known for `linexp`, we use a Newton approximation.

**Usage**

```r
t50(x)
```

**Arguments**

- `x` Result of a nlme fit, with named components `tempt`, `beta`, `logbeta`, `kappa`, `logkappa` depending on model. Function used `logbeta` when it is present, in `x`, otherwise `beta`, and similar for `logkappa/kappa`.  

Value

Half-emptying time. Name of evaluated function is returned as attribute `fun`. Negative of slope is returned as attribute `slope`. 
Index

c coef.nlme_gastempt 2
c coef.stan_gastempt 3
gastemptfunc, 3
linexp, 11
linexp(gastemptfunc), 3
linexp_auc (gastemptfunc), 3
linexp_log (gastemptfunc), 3
linexp_slope (gastemptfunc), 3
nlme_gastempt, 2, 3, 5
nlmControl, 5
plot.nlme_gastempt, 6
plot.stan_gastempt, 7
powexp (gastemptfunc), 3
powexp_log (gastemptfunc), 3
powexp_slope (gastemptfunc), 3
run_shiny, 7
simulate_gastempt, 8
stan_gastempt, 9
stan_model_names, 11
t50, 11