Package: litterfitter (via r-universe)

November 23, 2026

Type Package

Title Fits a Collection of Curves to Single-Cohort Decomposition Data

Version 0.3.0

Date 2026-11-20

Description Fit different model forms to single-cohort litter decomposition data (mass remaining through time) using likelihood-based estimation. Models span simple empirical to process-motivated forms with differing numbers of free parameters. Provides parameter estimates, uncertainty, and tools for model comparison/selection. Based on Cornwell & Weedon (2013) <doi:10.1111/2041-210X.12138>.

Depends R (>= 4.1.0)

LazyData true

LazyDataCompression xz

Encoding UTF-8

Imports generics, stats, graphics, grDevices, utils

Suggests ggplot2, vdiffr, testthat (>= 3.0.0), knitr, rmarkdown

License MIT + file LICENSE

URL https://traitecoevo.github.io/litterfitter/, https://github.com/traitecoevo/litterfitter

VignetteBuilder knitr

BugReports https://github.com/traitecoevo/litterfitter/issues

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Will Cornwell [aut, cre] (ORCID: <https://orcid.org/0000-0003-4080-4073>), James Weedon [aut], Guofang Liu [ctb]

Maintainer Will Cornwell <wcornwell@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-11-23 03:48:48 UTC

RemoteUrl https://github.com/cran/litterfitter

RemoteRef HEAD

RemoteSha d126249585770d7c6f9095e694d60a92847b8a4d

Contents

as_litfit_list . . . 2
bootstrap_parameters . . . 3
elementfit_methods . . . 4
fit_element . . . 6
fit_litter . . . 7
fit_litter_all . . . 10
litfit_list_methods . . . 11
litfit_tidiers . . . 13
litterfitter . . . 14
litterfitter_autoplot . . . 15
logLik.litfit . . . 17
pineneedles . . . 17
plot.elementfit . . . 18
plot.litfit . . . 19
plot.litfit_bootstrap . . . 20
plot.litfit_list . . . 21
plot_multiple_fits . . . 22
predict.litfit . . . 23
steady_state . . . 24
summary.litfit_bootstrap . . . 25
tidy.litfit_bootstrap . . . 26
time_to_prop_mass_remaining . . . 27

Index **28**

`as_litfit_list` *Combine existing fits into a litfit_list*

Description

Combine existing fits into a litfit_list

Usage

```
as_litfit_list(...)
```

Arguments

...	`litfit` objects, or a single list of them. Names are used as fit names; unnamed fits are named by their model (made unique if needed).

Details

Information criteria are only comparable between fits to the same data, so a warning is given if the fits were made to different data.

Value

A `litfit_list`.

Examples

```
f1 <- fit_litter(pineneedles$Year, pineneedles$Mass.remaining, "neg.exp",
  iters = 50, quiet = TRUE
)
f2 <- fit_litter(pineneedles$Year, pineneedles$Mass.remaining, "weibull",
  iters = 50, quiet = TRUE
)
fits <- as_litfit_list(f1, f2)
glance(fits)
```

`bootstrap_parameters` *Create a bootstrap distribution of a particular coefficient from a model fit*

Description

Create a bootstrap distribution of a particular coefficient from a model fit

Usage

```
bootstrap_parameters(x,nboot,upper,lower,...)
```

Arguments

x	an object of class "`litfit`" (from `fit_litter`) or "`elementfit`" (from `fit_element`)
nboot	number of bootstrap replications
upper	optional vector of upper bounds for the bootstrap replicates
lower	optional vector of lower bounds for the bootstrap replicates
...	passed to `optim`

Value

a `litfit_bootstrap` object: a matrix with one row per bootstrap replicate and one column per parameter, plus derived quantities: the implied steady state for a `litfit`, or the peak element remaining and the mass remaining at the peak for an `elementfit`. Failed replicates are recorded as `NA`.

Examples

```
fit <- fit_litter(
  time = pineneedles$Year,
  mass.remaining = pineneedles$Mass.remaining, model = "neg.exp", iters = 100
)
boot1 <- bootstrap_parameters(fit, nboot = 500)
```

`elementfit_methods` *Methods for elementfit objects*

Description

Methods for element fits from `fit_element`. The predictor is proportional mass remaining, so `newdata` gives mass remaining values.

Usage

```
## S3 method for class 'elementfit'
print(x, ...)

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

## S3 method for class 'elementfit'
fitted(object, ...)

## S3 method for class 'elementfit'
residuals(object, ...)

## S3 method for class 'elementfit'
nobs(object, ...)

## S3 method for class 'elementfit'
logLik(object, ...)

## S3 method for class 'elementfit'
predict(object, newdata = NULL, ...)

## S3 method for class 'elementfit'
```

```
tidy(x, boot = NULL, conf.level = 0.95, ...)

## S3 method for class 'elementfit'
glance(x, ...)

## S3 method for class 'elementfit'
augment(x, newdata = NULL, boot = NULL, conf.level = 0.95, ...)
```

Arguments

x, object	an elementfit object
...	unused
newdata	optional numeric vector of mass remaining values, or (for augment()) a data frame with a mass column
boot	optional litfit_bootstrap object from bootstrap_parameters for the same fit
conf.level	confidence level for intervals and bands

Details

glance() reports, besides information criteria, the peak proportional element remaining (peak_element) and the mass remaining at which it occurs (mass_at_peak). A peak of 1 at a mass of 1 means no net immobilisation.

Value

tidy(), glance() and augment() return a data.frame (see litfit_tidiers for the shared columns; augment() uses mass and element columns). plot() and print() return x invisibly.

Examples

```
fit <- fit_element(
  mass.remaining = pineneedles$Mass.remaining,
  element.remaining = c(1.1, 1.2, 1.3, 1, 0.8, 0.6),
  iters = 200, quiet = TRUE
)
coef(fit)
predict(fit, newdata = c(1, 0.75, 0.5, 0.25))
tidy(fit)
glance(fit)
head(augment(fit, newdata = seq(1, 0, by = -0.1)))
```

`fit_element` *Fit element (nutrient) dynamics during decomposition*

Description

Fits a curve describing how the proportion of an element remaining (for example nitrogen, N/N_0) changes with the proportion of litter mass remaining (M/M_0) in a single decomposition cohort. Elements are often immobilised early in decomposition (element remaining rises above 1) and released later.

Usage

```
fit_element(
  mass.remaining,
  element.remaining,
  model = c("import.model"),
  iters = 500,
  upper = NULL,
  lower = NULL,
  quiet = FALSE,
  ...
)
```

Arguments

`mass.remaining`	proportional mass remaining, M/M_0. Values slightly above 1 or below 0 are kept and treated as measurement error, with a warning (see `fit_litter`).
`element.remaining`	proportional element remaining, N/N_0. Values above 1 (net immobilisation) are expected.
`model`	the element model; currently only `"import.model"` (see Details)
`iters`	number of random starts for the fitting
`upper`, `lower`	optional bounds for the parameters, in the order given in Details
`quiet`	if `TRUE`, suppress the message reporting the number of successful fits
`...`	additional arguments passed to `optim`

Details

import.model is the nitrogen release model of Parton et al. (2007), written here for proportional mass remaining x:

$$f(x) = \frac{x/b}{\sqrt{(2ax/b)^2 + (1-(x/b)^2)^2}}, \qquad y = f(x)/f(1)$$

so the curve passes through 1 at $x = 1$ (no mass lost). b (bounds 0.01–1) sets the mass remaining around which the immobilisation peak is centred, and a (bounds 0.1–10) controls the height and

width of the peak; small a gives a tall, narrow peak. The original formulation used percent mass remaining (normalised at 100, with b between 1 and 100); this version is equivalent with b divided by 100.

Fitting uses the same multi-start likelihood approach as `fit_litter`, and gives the same warnings for parameters on a boundary and for mass values outside 0–1.

Value

An `elementfit` object, with methods for `print`, `plot`, `coef`, `fitted`, `residuals`, `predict`, `nobs`, `logLik` (so `AIC()` and `BIC()` work), `tidy`, `glance` and `augment`, and support in `bootstrap_parameters`. Its elements include `optimFit`, `logLik`, `fitAIC`, `fitAICc`, `fitBIC`, `mass`, `element`, `predicted`, `model` and `nparams`.

Author(s)

Baptiste Wijas, Will Cornwell

References

Parton, W., Silver, W. L., Burke, I. C., et al. (2007) Global-scale similarities in nitrogen release patterns during long-term decomposition. *Science*, 315, 361–364.

See Also

`fit_litter` for mass loss through time, `elementfit_methods`, `bootstrap_parameters`

Examples

```
fit <- fit_element(
  mass.remaining = pineneedles$Mass.remaining,
  element.remaining = c(1.1, 1.2, 1.3, 1, 0.8, 0.6),
  iters = 200, quiet = TRUE
)
fit
plot(fit)
glance(fit)
```

`fit_litter` *Fit different models to single cohort decomposition data*

Description

Non-linear fits of different models to the decomposition trajectory of one cohort (as in typical litterbag studies) data. Models range from very simple (and easy to fit with limited data) to more complex.

Usage

```
fit_litter(
  time,
  mass.remaining,
 model = c("neg.exp", "weibull", "discrete.parallel", "discrete.series", "cont.quality",
    "neg.exp.limit"),
  iters = 500,
  upper = NULL,
  lower = NULL,
  quiet = FALSE,
  ...
)
```

Arguments

time	time since decomposition began, that is, ti-t0
mass.remaining	proportional mass remaining, that is, mi/m0. Values above 1 (apparent mass gain, up to 2) or slightly below 0 (down to -0.1) are kept and treated as measurement error, with an informative warning (see Details).
model	there are six models currently implemented (see below); defaults to "neg.exp"
iters	Number of random starts for the fitting. Use higher numbers for models with larger numbers of parameters and for models that inherently tend to be less identifiable.
upper, lower	**Optional** user specified values for the upper and lower bounds used by optim in the fitting procedure. Use with care, only minimal sanity checking is currently implemented.
quiet	if TRUE, suppress the message reporting the number of successful fits
...	Additional arguments passed to optim

Details

the model likelihood is maximized using methods available in optim. Optimization methods to be used within optim can be specified through the control object (i.e., control$method). The default method is L-BFGS-B with bounds specific to each model. Each model

- **neg.exp** The classic negative exponential model–one parameter (Olson 1963)
- **weibull** The Weibull residence time model–two parameters (Frechet 1927)
- **discrete.parallel** Two pools in parallel with a term for the fraction of initial mass in each pool–three parameter (Manzoni et al. 2012)
- **discrete.series** A three parameter model in which there is the possibility of two sequential pools (Manzoni et al. 2012)
- **cont.quality** (Ågren and Bosatta 1996, see also Manzoni et al. 2012)
- **neg.exp.limit** A negative exponential decaying towards a non-zero asymptote ($Ae^{-kt} + B$)–three parameters

Warning: difficulty in finding the optimal solution is determined by an interaction between the nature and complexity of the likelihood space (which is both data- and model-dependent) as well as the optimization methods. There can never be a guarantee that the optimal solution is found, but using many random starting points will increase these odds. It should be noted that there is significant variation among models in identifiability, with some models inherently less identifiable likely due to a tendency to form for flat ridges in likelihood space. The confidence in the fit should be very low in these cases (see Cornwell and Weedon 2013). A number of random starting points are used in optimization and are given through the iters. The function checks whether the best 5 optimizations have converged on the same likelihood, and if they have not this function will return a warning.

Values outside 0–1: measurement error, especially early in decomposition or when decomposition is slow, can produce proportional mass remaining values slightly above 1 (i.e. negative mass loss). These values are kept in the fit, since removing or truncating them would bias the estimates, but `fit_litter` warns (class `litterfitter_mass_above_one`). If the data also show little overall mass loss, the warning notes that parameters may be poorly constrained and that derived quantities such as `time_to_prop_mass_remaining` and `steady_state` are extrapolations. Slightly negative values (down to -0.1) are similarly kept with a warning (class `litterfitter_mass_below_zero`). Use `suppressWarnings(..., classes = ...)` to silence these once checked.

Value

returns a litfit object with the following elements:

- **optimFit:** a list generated by the `optim` function
- **logLik:** the log-likelihood of the model
- **time:** vector of time (same as input)
- **mass:** vector of mass remaining (same as input)
- **predicted:** predicted values from the model for each of the points within time
- **model:** name of the model
- **nparams:** number of fit parameters in the model
- **AIC:** AIC of the model fit
- **AICc:** AICc of the model fit
- **BIC:** BIC of the model fit
- and some other potentially useful things

Author(s)

Will Cornwell and James Weedon

References

- Ågren, G. and Bosatta, E. (1996) Quality: a bridge between theory and experiment in soil organic matter studies. Oikos, 76, 522–528.
- Cornwell, W. K., and J. T. Weedon. (2013). Decomposition trajectories of diverse litter types: a model selection analysis. Methods in Ecology and Evolution.

- Frechet, M. (1927) Sur la loi de probabilite de lecart maximum. Ann de la Soc polonaise de Math, 6, 93–116.
- Manzoni, S., Pineiro, G., Jackson, R. B., Jobbagy, E. G., Kim, J. H., & Porporato, A. (2012). Analytical models of soil and litter decomposition: Solutions for mass loss and time-dependent decay rates. Soil Biology and Biochemistry, 50, 66-76.
- Olson, J.S. (1963) Energy storage and the balance of producers and decomposers in ecological systems. Ecology, 44, 322–331.

See Also

optim, steady_state, plot.litfit

Examples

```
data(pineneedles)
fit <- fit_litter(
  time = pineneedles$Year, mass.remaining = pineneedles$Mass.remaining,
  model = "neg.exp", iters = 1000
)
```

---

fit_litter_all *Fit several decomposition models to the same data*

Description

Fits each of the requested models with fit_litter and returns them together as a litfit_list, ready for model comparison with glance, tidy or augment.

Usage

```
fit_litter_all(
  time,
  mass.remaining,
  models = c("neg.exp", "weibull", "discrete.parallel", "discrete.series",
    "cont.quality", "neg.exp.limit"),
  iters = 500,
  ...
)
```

Arguments

time	time since decomposition began
mass.remaining	proportional mass remaining
models	models to fit (see fit_litter); defaults to all six
iters	number of random starts for each model
...	further arguments passed to fit_litter, such as upper, lower or optim control

Details

Warnings about the data (for example values above 1, or missing values removed) are given once rather than once per model. Warnings specific to one model, such as parameters on a boundary, are prefixed with the model name. A model that cannot be fitted (for example, too few observations for its number of parameters) is dropped with a warning; an error is raised only if no model can be fitted.

Value

A `litfit_list`: a named list of `litfit` objects.

See Also

`as_litfit_list` to combine existing fits, `litfit_list_methods`

Examples

```
fits <- fit_litter_all(
  time = pineneedles$Year, mass.remaining = pineneedles$Mass.remaining,
  models = c("neg.exp", "weibull", "cont.quality"), iters = 100
)
fits
glance(fits)
```

---

litfit_list_methods *Methods for litfit_list objects*

Description

Model comparison tools for several fits to the same data, as returned by `fit_litter_all` or `as_litfit_list`.

Usage

```
## S3 method for class 'litfit_list'
print(x, ...)

## S3 method for class 'litfit_list'
glance(x, ...)

## S3 method for class 'litfit_list'
tidy(x, ...)

## S3 method for class 'litfit_list'
augment(x, newdata = NULL, boot = NULL, conf.level = 0.95, ...)

## S3 method for class 'litfit_list'
```

```
coef(object, ...)

## S3 method for class 'litfit_list'
x[i]
```

Arguments

x	a litfit_list
...	unused
newdata	optional times at which to predict (see augment.litfit)
boot	optional named list of litfit_bootstrap objects, one per fit (names matching the fits), to add uncertainty bands
conf.level	confidence level for bands (default 0.95)
object	a litfit_list
i	indices or names of fits to keep

Details

- glance(): one row per fit, sorted by AICc, with delta_AICc (difference from the best model).
- tidy(): parameter estimates for every fit.
- augment(): fitted values (or predictions at newdata) for every fit, stacked, for plotting.

Every table starts with a fit column holding the names of the list.

Value

glance(), tidy() and augment() return a data.frame; coef() returns a named list of parameter vectors; [returns a litfit_list.

Examples

```
fits <- fit_litter_all(
  time = pineneedles$Year, mass.remaining = pineneedles$Mass.remaining,
  models = c("neg.exp", "weibull"), iters = 100
)
glance(fits)
tidy(fits)
head(augment(fits, newdata = seq(0, 5, by = 0.5)))
coef(fits)
fits["weibull"]
```

litfit_tidiers *Tidy, glance and augment methods for litfit objects*

Description

These methods return the results of a `fit_litter` fit as data frames, following the conventions of the **broom** package.

Usage

```
## S3 method for class 'litfit'
tidy(x, boot = NULL, conf.level = 0.95, ...)

## S3 method for class 'litfit'
glance(x, ...)

## S3 method for class 'litfit'
augment(x, newdata = NULL, boot = NULL, conf.level = 0.95, ...)
```

Arguments

`x`	a `litfit` object
`boot`	optional `litfit_bootstrap` object from `bootstrap_parameters` for the same fit. In `tidy()` it adds bootstrap standard errors and percentile intervals; in `augment()` it adds a pointwise percentile band for the fitted curve.
`conf.level`	confidence level for intervals and bands (default 0.95)
`...`	unused
`newdata`	optional numeric vector of times, or a data frame with a `time` column, at which to predict. If omitted, the original data are returned.

Details

- `tidy()`: one row per parameter.
- `glance()`: a one-row summary of the fit, including information criteria and the derived quantities time to 50\
- `augment()`: the data with fitted values and residuals, or predictions at new time points, optionally with a bootstrap uncertainty band.

Value

A `data.frame`.

`tidy()` `term`, `estimate`; with `boot` also `std.error`, `conf.low`, `conf.high`.

`glance()` `model`, `nobs`, `nparams`, `sigma` (residual standard deviation used in the likelihood), `logLik`, `AIC`, `AICc`, `BIC`, `converged`, `time_to_50`, `steady_state`. `time_to_50` is `NA` if the curve never reaches 50\ `steady_state` can be `Inf` (see `steady_state`).

augment() without newdata: time, mass, .fitted, .resid. With newdata: the columns of newdata (or time) plus .fitted. With boot: also .lower and .upper.

See Also

fit_litter, bootstrap_parameters, fit_litter_all

Examples

```
fit <- fit_litter(
  time = pineneedles$Year, mass.remaining = pineneedles$Mass.remaining,
  model = "weibull", iters = 100, quiet = TRUE
)
tidy(fit)
glance(fit)
augment(fit)

boot <- bootstrap_parameters(fit, nboot = 100)
tidy(fit, boot = boot)
head(augment(fit, newdata = seq(0, 5, by = 0.5), boot = boot))
```

---

litterfitter *litterfitter: methods for fitting curves to litter decomposition trajectories*

Description

There is a long tradition of studying the flux of carbon from the biosphere to the atmosphere by following a particular cohort of litter (wood, leaves, roots, or other organic material) through time. The resulting data are mass remaining and time. A variety of functional forms may be used to fit the resulting data. Some work better empirically. Some are better connected to a process-based understanding. Some have a small number of free parameters; others have more. This package matches decomposition data to a set of these curves using likelihood–based fitting.

Author(s)

Maintainer: Will Cornwell <wcornwell@gmail.com> (ORCID)

Authors:

- Will Cornwell <wcornwell@gmail.com> (ORCID)
- James Weedon

Other contributors:

- Guofang Liu [contributor]

References

Cornwell, W. K., and J. T. Weedon. (2013). Decomposition trajectories of diverse litter types: a model selection analysis. Methods in Ecology and Evolution.

Manzoni, S., Pineiro, G., Jackson, R. B., Jobbagy, E. G., Kim, J. H., & Porporato, A. (2012). Analytical models of soil and litter decomposition: Solutions for mass loss and time-dependent decay rates. Soil Biology and Biochemistry, 50, 66-76.

See Also

Useful links:

- https://traitecoevo.github.io/litterfitter/
- https://github.com/traitecoevo/litterfitter
- Report bugs at https://github.com/traitecoevo/litterfitter/issues

---

`litterfitter_autoplot` *ggplot2 plots of litterfitter fits*

Description

`autoplot()` methods that draw fits, model comparisons and bootstrap distributions with **ggplot2**. They require ggplot2 to be installed and loaded (`library(ggplot2)`); the result is a regular ggplot object that can be modified with further layers, scales and themes.

Usage

```
## S3 method for class 'litfit'
autoplot(object, boot = NULL, conf.level = 0.95, equation = TRUE, n = 200, ...)

## S3 method for class 'elementfit'
autoplot(object, boot = NULL, conf.level = 0.95, equation = TRUE, n = 200, ...)

## S3 method for class 'litfit_list'
autoplot(object, boot = NULL, conf.level = 0.95, n = 200, ...)

## S3 method for class 'litfit_bootstrap'
autoplot(object, terms = NULL, conf.level = 0.95, ...)
```

Arguments

`object`	a `litfit`, `elementfit`, `litfit_list` or `litfit_bootstrap` object
`boot`	optional `litfit_bootstrap` from `bootstrap_parameters` for the same fit (for a `litfit_list`, a named list of them), to draw an uncertainty band
`conf.level`	confidence level for bands and shaded intervals (default 0.95)
`equation`	if `TRUE`, show the fitted equation as the subtitle

n	number of points used to draw curves
...	unused
terms	for bootstraps, the parameters or derived quantities to show (default all)

Details

- `litfit`: the data, fitted curve and (optionally) a bootstrap band, with the fitted equation as the subtitle.
- `elementfit`: as above, with mass remaining decreasing from left to right and a dashed line at the initial element amount.
- `litfit_list`: the curves of all models, coloured with the Okabe-Ito palette and labelled with their difference in AICc from the best model ("delta AICc").
- `litfit_bootstrap`: the bootstrap distribution of each parameter and derived quantity in its own panel, with the central interval shaded and lines at the mean (red) and median (blue).

Value

a `ggplot` object

See Also

`augment.litfit` and `tidy.litfit_bootstrap`, which return the data behind these plots for building custom figures; `plot.litfit` for base graphics

Examples

```
library(ggplot2)
fit <- fit_litter(
  time = pineneedles$Year, mass.remaining = pineneedles$Mass.remaining,
  model = "weibull", iters = 100, quiet = TRUE
)
boot <- bootstrap_parameters(fit, nboot = 200)
autoplot(fit, boot = boot)
autoplot(boot)

fits <- fit_litter_all(pineneedles$Year, pineneedles$Mass.remaining,
  models = c("neg.exp", "weibull", "cont.quality"), iters = 100
)
autoplot(fits) + theme_bw()
```

logLik.litfit *Log-likelihood of a litfit object*

Description

Returns the log-likelihood of the fitted model, which allows the standard AIC and BIC functions to be used to compare several litfit objects, e.g. AIC(fit1, fit2).

Usage

```
## S3 method for class 'litfit'
logLik(object, ...)
```

Arguments

object	a litfit object
...	unused

Details

The degrees of freedom are the number of fitted curve parameters (the residual standard deviation is not counted), consistent with the fitAIC, fitAICc and fitBIC values stored in the object and with Cornwell & Weedon (2013).

Value

an object of class logLik

pineneedles *decomposition trajectory for pine needles*

Description

data from Hobbie et al means of pine needle decomposition

Usage

```
data(pineneedles)
```

Format

a data.frame with two columns

Author(s)

Will Cornwell

Source

Hobbie et al.

`plot.elementfit` *Plot an element fit*

Description

Plots proportional element remaining against proportional mass remaining, with mass remaining decreasing from left to right so that decomposition proceeds along the x axis. A dashed line marks the initial amount of the element.

Usage

```
## S3 method for class 'elementfit'
plot(x, formulae.cex = 1, boot = NULL, conf.level = 0.95, ...)
```

Arguments

`x`	an `elementfit` object from `fit_element`
`formulae.cex`	size of the parameter text; use `NA` to omit it
`boot`	optional `litfit_bootstrap` object for the same fit, to draw an uncertainty band
`conf.level`	confidence level for the band (default 0.95)
`...`	additional arguments passed to `plot.default`

Value

`x`, invisibly

See Also

`fit_element`

Examples

```
fit <- fit_element(
  mass.remaining = pineneedles$Mass.remaining,
  element.remaining = c(1.1, 1.2, 1.3, 1, 0.8, 0.6),
  iters = 200, quiet = TRUE
)
plot(fit)
```

`plot.litfit` *Plot a decomposition trajectory and its fitted curve*

Description

Plots the data and fitted curve from a `litfit` object, with the fitted equation. If a bootstrap is supplied, a pointwise percentile band for the curve is added.

Usage

```
## S3 method for class 'litfit'
plot(x, formulae.cex = 1, boot = NULL, conf.level = 0.95, ...)
```

Arguments

`x`	a `litfit` object from `fit_litter`
`formulae.cex`	size of the equation text; use `NA` to omit the equation
`boot`	optional `litfit_bootstrap` object from `bootstrap_parameters` for the same fit, to draw an uncertainty band
`conf.level`	confidence level for the band (default 0.95)
`...`	additional arguments passed to `plot.default`, which override the defaults (e.g. `xlab`, `ylab`, `main`, `pch`)

Value

`x`, invisibly

Author(s)

Will Cornwell

See Also

`fit_litter`, `plot.litfit_list`; `autoplot()` methods for ggplot2 output (`litterfitter_autoplot`)

Examples

```
fit <- fit_litter(
  time = pineneedles$Year, mass.remaining = pineneedles$Mass.remaining,
  model = "weibull", iters = 100, quiet = TRUE
)
plot(fit)

boot <- bootstrap_parameters(fit, nboot = 100)
plot(fit, boot = boot, main = "Weibull fit with 95% band")
```

`plot.litfit_bootstrap` *Plot bootstrap distributions*

Description

Plots the density of the bootstrap distribution of one or more parameters (or derived quantities) from `bootstrap_parameters`.

Usage

```
## S3 method for class 'litfit_bootstrap'
plot(x, coef.index = 1, bw = "nrd0", conf.level = 0.95, ...)
```

Arguments

`x`	a `litfit_bootstrap` object
`coef.index`	which column(s) to plot: a number, a column name such as `"k"` or `"steady_state"`, a vector of either, or `"all"`. Several columns are drawn in a grid of panels. With `"all"`, columns without at least two finite values (e.g. an always-infinite steady state) are skipped with a warning.
`bw`	bandwidth (or bandwidth rule, see `density`) for the density
`conf.level`	width of the shaded central interval (default 0.95)
`...`	additional arguments passed to `plot.default`

Details

The grey fill covers the central `conf.level` interval of the distribution. The red dashed line shows the mean and the blue dashed line the median. Non-finite values (e.g. an infinite steady state) and failed replicates are ignored.

Value

`x`, invisibly

Author(s)

James Weedon

See Also

`bootstrap_parameters`, `density`

Examples

```
fit <- fit_litter(
  time = pineneedles$Year, mass.remaining = pineneedles$Mass.remaining,
  model = "weibull", iters = 100, quiet = TRUE
)
boot <- bootstrap_parameters(fit, nboot = 200)
plot(boot)
plot(boot, coef.index = "all")
```

`plot.litfit_list` *Plot several fits to the same data*

Description

Plots the data with the fitted curve of every model in a `litfit_list`, and a legend table of AIC, AICc and BIC sorted by AICc.

Usage

```
## S3 method for class 'litfit_list'
plot(
  x,
  col = NULL,
  legend.pos = "topright",
  legend.bty = "o",
  legend.cex = 0.8,
  ...
)
```

Arguments

`x`	a `litfit_list` from `fit_litter_all` or `as_litfit_list`
`col`	colours for the curves, recycled; defaults to the colour-blind safe Okabe-Ito palette
`legend.pos`	position of the legend, passed to `legend`; use `NA` to omit it
`legend.bty`	box type of the legend (`"o"` or `"n"`)
`legend.cex`	size of the legend text
`...`	additional arguments passed to `plot.default`

Details

Data points are taken from the first fit; fits in a `litfit_list` are normally made to the same data.

Value

`x`, invisibly

See Also

plot_multiple_fits, glance.litfit_list

Examples

```
fits <- fit_litter_all(
  time = pineneedles$Year, mass.remaining = pineneedles$Mass.remaining,
  models = c("neg.exp", "weibull", "cont.quality"), iters = 100
)
plot(fits)
```

---

plot_multiple_fits *Fit and plot several models on one graph with model selection results*

Description

Fits several models to the same data with fit_litter_all and plots them together with plot.litfit_list. Existing fits can be plotted instead by passing a litfit_list (or a list of litfit objects) as the first argument, which avoids refitting.

Usage

```
plot_multiple_fits(
  time,
  mass.remaining,
 model = c("neg.exp", "weibull", "discrete.parallel", "discrete.series", "cont.quality",
    "neg.exp.limit"),
  color = NULL,
  iters = 500,
  bty = "o",
  ...
)
```

Arguments

time	vector of time points, or a litfit_list / list of litfit objects to plot without refitting
mass.remaining	vector of mass remaining (ignored when time is a list of fits)
model	vector of models to fit and plot (see fit_litter)
color	colours for the curves; defaults to the Okabe-Ito palette
iters	number of random starts, passed to fit_litter
bty	box type of the legend
...	additional arguments passed to plot.default

Value

the fits, invisibly, as a `litfit_list`

Author(s)

Liu Guofang

See Also

`fit_litter_all`, `plot.litfit_list`

Examples

```
plot_multiple_fits(
  time = pineneedles$Year,
  mass.remaining = pineneedles$Mass.remaining,
  bty = "n", model = c("neg.exp", "weibull"), iters = 200
)

# plot existing fits without refitting
fits <- fit_litter_all(pineneedles$Year, pineneedles$Mass.remaining,
  models = c("neg.exp", "cont.quality"), iters = 100
)
plot_multiple_fits(fits)
```

`predict.litfit` *Predict method for litfit objects*

Description

Generated predicted values for (new) time points from a litfit model fit

Usage

```
## S3 method for class 'litfit'
predict(object,newdata=NULL,...)
```

Arguments

`object`	litfit object
`newdata`	optional vector of new Time points at which to predict mass remaining. If not specified, Time points from the original fit are used.
`...`	further arguments passed to or from other methods.

Details

to do

Value

predicted values from a litfit object

Author(s)

Will Cornwell

James Weedon

See Also

fit_litter

Examples

```
fit <- fit_litter(
  time = c(0, 1, 2, 3, 4, 5, 6), mass.remaining = c(1, 0.9, 1.01, 0.4, 0.6, 0.2, 0.01),
  "neg.exp", iters = 250
)
predict(fit, newdata = 1:10)
```

---

steady_state *Estimate Steady State Biomass*

Description

Estimate Steady State Biomass

Usage

```
steady_state(x = NULL, pars = NULL, model = NULL)
```

Arguments

x	A litfit object. If provided, pars and model parameters are extracted from this object.
pars	A numeric vector of parameters for the model. Only needed if x is not provided.
model	A character string specifying the decomposition model. Must be one of the following: "neg.exp", "weibull", "discrete.parallel", "discrete.series", "cont.quality" or "neg.exp.limit". Only needed if x is not provided.

Details

Computes the steady state biomass, as a proportion of the annual input, based on a given model fit or parameters.

For neg.exp.limit, a fraction B of the litter never decomposes, so the implied steady state is infinite whenever $B > 0$. Because B is bounded just above zero during fitting, values of $B < 10^{-4}$ are treated as zero and the steady state is A/k.

Value

A named numeric value representing the estimated steady state biomass from the specified model.

Author(s)

Will Cornwell

See Also

`fit_litter` for generating `litfit` objects.

Examples

```
# Example with litfit object
fit <- fit_litter(
  time = c(0, 1, 2, 3, 4, 5, 6),
  mass.remaining = c(1, 0.9, 1.01, 0.4, 0.6, 0.2, 0.01),
  model = "neg.exp",
  iters = 250
)
steady_state(fit)

# Example with specific model and parameter values
steady_state(pars = c(6, 2), model = "weibull")
```

---

summary.litfit_bootstrap

Summarise a bootstrap distribution of parameters from a litfit object

---

Description

Summarise a bootstrap distribution of parameters from a litfit object

Usage

```
## S3 method for class 'litfit_bootstrap'
summary(object, probs = c(0.025, 0.5, 0.975), ...)
```

Arguments

`object`	a `litfit_bootstrap` object from `bootstrap_parameters`
`probs`	quantiles to report (default gives a 95 percent interval and the median)
`...`	unused

Value

a matrix with one row per parameter (plus steady state) giving the mean and the requested quantiles of the bootstrap distribution, ignoring failed replicates

Examples

```
fit <- fit_litter(
  time = pineneedles$Year,
  mass.remaining = pineneedles$Mass.remaining, model = "neg.exp", iters = 100
)
boot1 <- bootstrap_parameters(fit, nboot = 200)
summary(boot1)
```

`tidy.litfit_bootstrap` *Tidy a bootstrap distribution*

Description

Returns the bootstrap replicates from `bootstrap_parameters` in long format, one row per replicate and parameter (including the implied steady state). Failed replicates are dropped.

Usage

```
## S3 method for class 'litfit_bootstrap'
tidy(x, ...)
```

Arguments

`x`	a `litfit_bootstrap` object
`...`	unused

Value

A `data.frame` with columns `replicate`, `term` and `value`.

Examples

```
fit <- fit_litter(
  time = pineneedles$Year, mass.remaining = pineneedles$Mass.remaining,
  model = "neg.exp", iters = 50, quiet = TRUE
)
boot <- bootstrap_parameters(fit, nboot = 50)
head(tidy(boot))
```

`time_to_prop_mass_remaining` *Get the predicted time until half mass loss for a litter decomposition trajectory*

Description

Get estimated time to 0.5 (or an alternate threshold) mass loss from a particular fit to a litter decomposition trajectory

Usage

```
time_to_prop_mass_remaining(x,threshold.mass=0.5)
```

Arguments

`x`	a litfit object
`threshold.mass`	mass loss threshold in proportion, default is 0.5

Details

this function finds the time to a specified mass loss percentage

Value

numeric value that represents time to a specified mass loss percentage

Author(s)

Will Cornwell

See Also

`fit_litter` `plot.litfit`

Examples

```
fit <- fit_litter(
  time = pineneedles$Year, mass.remaining = pineneedles$Mass.remaining,
  model = "neg.exp", iters = 1000
)
time_to_prop_mass_remaining(fit, threshold.mass = 0.5)
```

Index

∗ **datasets**
    pineneedles, 17
[.litfit_list *(*litfit_list_methods*)*, 11

AIC, *17*
as_litfit_list, 2, *11*, *21*
augment, *10*
augment.elementfit
        *(*elementfit_methods*)*, 4
augment.litfit, *12*, *16*
augment.litfit *(*litfit_tidiers*)*, 13
augment.litfit_list
        *(*litfit_list_methods*)*, 11
autoplot.elementfit
        *(*litterfitter_autoplot*)*, 15
autoplot.litfit
        *(*litterfitter_autoplot*)*, 15
autoplot.litfit_bootstrap
        *(*litterfitter_autoplot*)*, 15
autoplot.litfit_list
        *(*litterfitter_autoplot*)*, 15

BIC, *17*
bootstrap_parameters, 3, *5*, *7*, *13–15*, *19*, *20*, *25*, *26*

coef.elementfit *(*elementfit_methods*)*, 4
coef.litfit_list *(*litfit_list_methods*)*, 11

density, *20*

elementfit_methods, 4, *7*

fit_element, *3*, *4*, 6, *18*
fit_litter, *3*, *6*, 7, *7*, *10*, *13*, *14*, *19*, *22*, *24*, *25*, *27*
fit_litter_all, 10, *11*, *14*, *21–23*
fitted.elementfit *(*elementfit_methods*)*, 4

glance, *10*
glance.elementfit *(*elementfit_methods*)*, 4
glance.litfit *(*litfit_tidiers*)*, 13
glance.litfit_list, *22*
glance.litfit_list
        *(*litfit_list_methods*)*, 11

legend, *21*
litfit_list_methods, 11, *11*
litfit_tidiers, *5*, 13
litterfitter, 14
litterfitter-package *(*litterfitter*)*, 14
litterfitter_autoplot, 15, *19*
logLik.elementfit *(*elementfit_methods*)*, 4
logLik.litfit, 17

nobs.elementfit *(*elementfit_methods*)*, 4

optim, *3*, *6*, *8–10*

pineneedles, 17
plot.default, *18–22*
plot.elementfit, 18
plot.litfit, *10*, *16*, 19, *27*
plot.litfit_bootstrap, 20
plot.litfit_list, *19*, 21, *22*, *23*
plot_multiple_fits, 22, *22*
predict.elementfit
        *(*elementfit_methods*)*, 4
predict.litfit, 23
print.elementfit *(*elementfit_methods*)*, 4
print.litfit_list
        *(*litfit_list_methods*)*, 11

residuals.elementfit
        *(*elementfit_methods*)*, 4

steady_state, *9*, *10*, *13*, 24
summary.litfit_bootstrap, 25

tidy, *10*
tidy.elementfit *(*elementfit_methods*)*, 4
tidy.litfit *(*litfit_tidiers*)*, 13
tidy.litfit_bootstrap, *16*, 26
tidy.litfit_list *(*litfit_list_methods*)*, 11
time_to_prop_mass_remaining, *9*, 27