Package ‘morse’

August 29, 2016

Type Package

Title MOdelling Tools for Reproduction and Survival Data in Ecotoxicology

Version 2.2.0

Date 2016-06-03

Author Marie Laure Delignette-Muller [aut], Philippe Ruiz [aut], Sandrine Charles [aut], Wandrille Duchemin [ctb], Christelle Lopes [ctb], Guillaume Kon-Kam-king [ctb], Philippe Veber [aut]

Maintainer Philippe Veber <philippe.veber@univ-lyon1.fr>

Description Tools for ecotoxicologists and regulators dedicated to the mathematical and statistical modelling of bioassay data. They use advanced and innovative methods for a valuable quantitative environmental risk assessment.

Depends R (>= 3.0.0)

SystemRequirements jags (>= 4.0.0) (see http://mcmc-jags.sourceforge.net)

Imports coda, dplyr, epitools, graphics, grDevices, ggplot2 (>= 2.1.0), grid, gridExtra, methods, reshape2, rjags (>= 4.0), stats, stringr

License GPL (>= 2)

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

RoxygenNote 5.0.1

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2016-06-08 15:40:21
**R topics documented:**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>morse-package</td>
<td>3</td>
</tr>
<tr>
<td>as.survData</td>
<td>4</td>
</tr>
<tr>
<td>cadmium1</td>
<td>5</td>
</tr>
<tr>
<td>cadmium2</td>
<td>5</td>
</tr>
<tr>
<td>chlordan</td>
<td>6</td>
</tr>
<tr>
<td>copper</td>
<td>7</td>
</tr>
<tr>
<td>dichromate</td>
<td>8</td>
</tr>
<tr>
<td>plot.reproData</td>
<td>9</td>
</tr>
<tr>
<td>plot.reproFitTT</td>
<td>10</td>
</tr>
<tr>
<td>plot.survData</td>
<td>11</td>
</tr>
<tr>
<td>plot.survFitTKTD</td>
<td>13</td>
</tr>
<tr>
<td>plot.survFitTT</td>
<td>14</td>
</tr>
<tr>
<td>plotDoseResponse</td>
<td>16</td>
</tr>
<tr>
<td>plotDoseResponse.reproData</td>
<td>16</td>
</tr>
<tr>
<td>plotDoseResponse.survData</td>
<td>18</td>
</tr>
<tr>
<td>ppc</td>
<td>19</td>
</tr>
<tr>
<td>ppc.reproFitTT</td>
<td>19</td>
</tr>
<tr>
<td>ppc.survFitTKTD</td>
<td>20</td>
</tr>
<tr>
<td>ppc.survFitTT</td>
<td>21</td>
</tr>
<tr>
<td>print.reproFitTT</td>
<td>22</td>
</tr>
<tr>
<td>print.survFitTKTD</td>
<td>23</td>
</tr>
<tr>
<td>print.survFitTT</td>
<td>24</td>
</tr>
<tr>
<td>propiconazole</td>
<td>25</td>
</tr>
<tr>
<td>reproData</td>
<td>26</td>
</tr>
<tr>
<td>reproDataCheck</td>
<td>27</td>
</tr>
<tr>
<td>reproFitTT</td>
<td>28</td>
</tr>
<tr>
<td>summary.reproData</td>
<td>30</td>
</tr>
<tr>
<td>summary.reproFitTT</td>
<td>30</td>
</tr>
<tr>
<td>summary.survData</td>
<td>31</td>
</tr>
<tr>
<td>summary.survFitTKTD</td>
<td>32</td>
</tr>
<tr>
<td>summary.survFitTT</td>
<td>33</td>
</tr>
<tr>
<td>survData</td>
<td>34</td>
</tr>
<tr>
<td>survDataCheck</td>
<td>35</td>
</tr>
<tr>
<td>survFitTKTD</td>
<td>36</td>
</tr>
<tr>
<td>survFitTT</td>
<td>38</td>
</tr>
<tr>
<td>zinc</td>
<td>39</td>
</tr>
</tbody>
</table>

**Index** 41
Description

Provides tools for the analysis of survival/reproduction bioassay data in quantitative environmental risk assessment. It can be used to explore/visualize experimental data, and to perform an estimation of \( LC_x \) (Lethal Concentration for x\% of individuals) or \( EC_x \) (x\% Effect Concentration) values by fitting exposure-response curves. The \( LC_x/EC_x \) and parameters of the curve are provided along with an indication of the uncertainty of the estimation. It can be used to perform an estimation of \( NEC \) (No Effect Concentration) value by fitting a toxico-kinetic toxico-dynamic model.

Details

Estimation procedures in MORSE can be used without a deep knowledge of their underlying probabilistic model or inference methods. Rather, they were designed to behave as well as possible without requiring a user to provide values for some obscure parameters. That said, MORSE models can also be used as a first step to tailor new models for more specific situations.

The package currently handles survival and reproduction data. Functions dedicated to survival (resp. reproduction) analysis start with a \texttt{surv} (resp. \texttt{repro}) prefix. MORSE provides a similar workflow in both cases:

1. create and validate a dataset
2. explore a dataset
3. plot a dataset
4. fit a model on a dataset and output the expected estimates

Those steps are presented in more details in the "Tutorial" vignette, while a more formal description of the estimation procedures are provided in the vignette called "Models in MORSE package". Please refer to these documents for further introduction to the use of MORSE.

This reference manual is a detailed description of the functions exposed in the package.

Getting started The package uses the \texttt{rjags} package (Plummer, 2013), an R interface to the JAGS library for Bayesian model estimation. Note that the \texttt{rjags} package does not include a copy of the JAGS library: you need to install it separately. For instructions on downloading JAGS, see the home page at \url{http://mcmc-jags.sourceforge.net}. Once done, simply follow the steps described in the tutorial vignette.

```
Package: morse
Type: Package
Version: 2.2.0
Date: 2016-03-03
License: GPL (>=2)
```
as.survData

Author(s)
Marie Laure Delignette-Muller <marielaure.delignettemuller@vetagro-sup.fr>, Philippe Ruiz <philippe.ruiz@univ-lyon1.fr>, Sandrine Charles <sandrine.charles@univ-lyon1.fr>, Wandrille Duchemin <wandrille.duchemin@insa-lyon.fr>, Christelle Lopes <christelle.lopes@univ-lyon1.fr>, Guillaume Kon-Kam-king <guillaume.kon-kam-king@univ-lyon1.fr>, Christelle Lopes <christelle.lopes@univ-lyon1.fr>, Guillaume Kon-Kam-king <guillaume.kon-kam-king@univ-lyon1.fr>, Philippe Veber <philippe.veber@univ-lyon1.fr>
Maintainer: Philippe Ruiz <philippe.ruiz@univ-lyon1.fr>

References

See Also
rjags, ggplot

as.survData  Cast for classes derived from survData

Description
Use this function to access survData methods on an object of a derived class (e.g. reproData)

Usage
as.survData(x)

Arguments
x an S3 object of a class derived from survData

See Also
reproData and survData

Examples
data(zinc)
x <- reproData(zinc)

# Compare
plot(x)

#and
plot(as.survData(x))
**Description**

Reproduction and survival datasets of chronic laboratory bioassays with *Daphnia magna* freshwater invertebrate exposed to five concentrations of one metal contaminant (cadmium) during 21 days. Five concentrations were tested, with four replicates per concentration. Each replicate contained 10 organisms. Reproduction and survival were monitored at 10 time points.

**Usage**

```r
data(cadmium1)
```

**Format**

A data frame with 200 observations of the following five variables:

- **replicate**: A vector of class `integer` with the replicate code (Q to T).
- **conc**: A vector of class `numeric` with the cadmium concentrations in µg.L⁻¹.
- **time**: A vector of class `integer` with the time points (in days from the beginning of the experiment $t = 0$).
- **nsurv**: A vector of class `integer` with the number of alive individuals at each time point for each concentration and each replicate.
- **nrepro**: A vector of class `integer` with the number of offspring at each time point for each concentration and each replicate.

**References**


---

**cadmium2**  
*Reproduction and survival datasets for snails exposed to cadmium during 56 days*

**Description**

Reproduction and survival datasets of chronic laboratory bioassays with snails exposed to six concentrations of one metal contaminant (cadmium) during 56 days. Six concentrations were tested, with six replicates per concentration. Each replicate contained five organisms. Reproduction and survival were monitored at 17 time points.
Usage
data(cadmium2)

Format
A data frame with 612 observations of the following five variables:

replicate A vector of class factor with the replicate code (A to F).
conc A vector of class integer with the cadmium concentrations in $\mu g.L^{-1}$.
time A vector of class integer with the time points (in days from the beginning of the experiment $t = 0$).
Nsurv A vector of class integer with the number of alive individuals at each time point for each concentration and each replicate.
Nrepro A vector of class integer with the number of offspring at each time point for each concentration and each replicate.

References
Format

A data frame with 1320 observations of the following five variables:

- `replicate` A vector of class `integer` with the replicate code (1 to 10).
- `conc` A vector of class `numeric` with the chlordan concentrations in $\mu g.L^{-1}$.
- `time` A vector of class `integer` with the time points (in days from the beginning of the experiment $t = 0$).
- `Nsurv` A vector of class `integer` with the number of alive individuals at each time point for each concentration and each replicate.
- `Nrepro` A vector of class `integer` with the number of offspring at each time point for each concentration and each replicate.

References


Description

Reproduction and survival datasets of chronic laboratory bioassays with *Daphnia magna* freshwater invertebrate exposed to five concentrations of one metal contaminant (copper) during 21 days. Five concentrations were tested, with three replicates per concentration. Each replicate contained 20 organisms. Reproduction and survival were monitored at 16 time points.

Usage

data(copper)

Format

A data frame with 240 observations of the following five variables:

- `replicate` A vector of class `factor` with the replicate code (A to C).
- `conc` A vector of class `numeric` with the copper concentrations in $\mu g.L^{-1}$.
- `time` A vector of class `integer` with the time points (in days from the beginning of the experiment $t = 0$).
- `Nsurv` A vector of class `integer` with the number of alive individuals at each time point for each concentration and each replicate.
- `Nrepro` A vector of class `integer` with the number of offspring at each time point for each concentration and each replicate.
References


---

dichromate

Survival datasets for *Daphnia magna* exposed to dichromate during 21 days

Description

Survival datasets of chronic laboratory bioassays with *Daphnia magna* freshwater invertebrate exposed to six concentrations of one oxidizing agent (potassium dichromate) during 21 days. Six concentrations were tested with one replicate of 50 organisms per concentration. Survival is monitored at 10 time points.

Usage

data(dichromate)

Format

A data frame with 60 observations on the following four variables:

- **replicate** A vector of class factor with the replicate code (1).
- **conc** A vector of class numeric with dichromate concentrations in \text{mg.L}^{-1}.
- **time** A vector of class integer with the time points (in days from the beginning of the experiment \(t = 0\)).
- **nsurv** A vector of class integer with the number of alive individuals at each time point for each concentration and each replicate.

References

**plot.reproData**  

*Plotting method for reproData objects*

**Description**

This is the generic plot S3 method for the reproData class. It plots the cumulated number of offspring as a function of time.

**Usage**

```r
## S3 method for class 'reproData'
plot(x, xlab, ylab = "Cumulated Number of offspring",
     main = NULL, concentration = NULL, style = "generic",
     pool.replicate = FALSE, addlegend = FALSE, remove.someLabels = FALSE,
     ...)  
```

**Arguments**

- `x`: an object of class reproData
- `xlab`: a title for the x-axis (optional)
- `ylab`: a title for the y-axis
- `main`: main title for the plot
- `concentration`: a numeric value corresponding to some concentration in data. If `concentration = NULL`, draws a plot for each concentration
- `style`: graphical backend, can be 'generic' or 'ggplot'
- `pool.replicate`: if TRUE, the datapoints of each replicate are summed for a same concentration
- `addlegend`: if TRUE, adds a default legend to the plot
- `remove.someLabels`: if TRUE, removes 3/4 of X-axis labels in 'ggplot' style to avoid the label overlap
- `...`: Further arguments to be passed to generic methods

**Note**

When `style = "ggplot"`, the function calls function `ggplot` and returns an object of class `ggplot`.

**Examples**

```r
library(ggplot2)

# (1) Load the data
data(cadmium1)
cadmium1 <- reproData(cadmium1)

# (2) Plot the reproduction data
```
plot(cadmium1)

# (3) Plot the reproduction data for a fixed concentration
plot(cadmium1, concentration = 4.36, style = "ggplot")

plot_reproFitTT

Plotting method for reproFitTT objects

Description

This is the generic plot S3 method for the reproFitTT class. It plots exposure-response fits for target time reproduction analysis.

Usage

```r
## S3 method for class 'reproFitTT'
plot(x, xlab = "Concentration",
     ylab = "Nb of offspring per ind.day", main = NULL, fitcol = "red",
     fitlty = 1, fitlwd = 1, spaghetti = FALSE, cicol = "pink!",
     cilty = 1, cilwd = 1, addlegend = FALSE, log.scale = FALSE,
     style = "generic", ...)
```

Arguments

- `x` an object of class reproFitTT
- `xlab` a title for the x-label
- `ylab` a title for the y-label
- `main` main title for the plot
- `fitcol` color used for the fitted curve
- `fitlty` line type for the fitted curve
- `fitlwd` width of the fitted curve
- `spaghetti` if TRUE, the credible interval is represented by multiple curves
- `cicol` color for the 95 % credible limits of the fitted curve
- `cilty` line type for the 95 % credible limits of the fitted curve
- `cilwd` width of the 95 % credible limits of the fitted curve
- `addlegend` if TRUE, adds a default legend to the plot
- `log.scale` if TRUE, displays x-axis in log-scale
- `style` graphical backend, can be 'generic' or 'ggplot'
- `...` Further arguments to be passed to generic methods
Details

The fitted curve represents the **estimated reproduction rate** after the target time has passed as a function of the concentration of pollutant. The function plots 95% credible intervals for the estimated reproduction rate (by default the red area around the fitted curve). Both types of intervals are taken at the same level. Typically a good fit is expected to display a large overlap between the two intervals. If spaghetti = TRUE, the credible intervals are represented by two dotted lines limiting the credible band, and a spaghetti plot is added to this band. It consists of the representation of simulated curves using parameter values sampled in the posterior distribution (10% of the MCMC chains are randomly taken for this sample).

Note

When style = "ggplot", the function calls function `ggplot` and returns an object of class `ggplot`.

Examples

```r
# (1) Load the data
data(cadmium1)

# (2) Create an object of class "reproData"
dat <- reproData(cadmium1)

## Not run:
# (3) Run the reproFitTT function with the log-logistic gamma-poisson model
out <- reproFitTT(dat, stoc.part = "gammapoisson",
                  ecx = c(5, 10, 15, 20, 30, 50, 80), quiet = TRUE)

# (4) Plot the fitted curve with ggplot style
plot(out, xlab = expression("Concentration in" ~ mu~g.L^(-1)),
     fitcol = "blue", cicol = "blue", style = "ggplot",
     main = "Log-logistic response to concentration")

## End(Not run)
```

plot.survData  
*Plotting method for survData objects*

Description

This is the generic plot S3 method for the survData class. It plots the number of survivors as a function of time.
Usage

```r
## S3 method for class 'survData'
plot(x, xlab, ylab = "Number of surviving individuals",
     main = NULL, concentration = NULL, style = "generic",
     pool.replicate = FALSE, addlegend = FALSE, remove.someLabels = FALSE,
     )
```

Arguments

- `x`: an object of class `survData`
- `xlab`: a title for the x-axis (optional)
- `ylab`: a label for the y-axis
- `main`: main title for the plot
- `concentration`: a numeric value corresponding to some concentration in data. If `concentration` = `NULL`, draws a plot for each concentration
- `style`: graphical backend, can be 'generic' or 'ggplot'
- `pool.replicate`: if `TRUE`, the datapoints of each replicate are summed for a same concentration
- `addlegend`: if `TRUE`, adds a default legend to the plot
- `remove.someLabels`: if `TRUE`, removes 3/4 of X-axis labels in 'ggplot' style to avoid label overlap

... Further arguments to be passed to generic methods

Note

When `style = "ggplot"`, the function calls function `ggplot` and returns an object of class `ggplot`.

Examples

```r
library(ggplot2)

# (1) Load the data
data(zinc)
zinc <- survData(zinc)

# (2) Plot survival data with a ggplot style
plot(zinc, style = "ggplot")

# (3) Plot the survival data for a fixed concentration
plot(zinc, concentration = 0.66, style = "ggplot")
```
Description

This is the generic plot S3 method for the survFitTKTD. It plots the fits obtained for each concentration of pollutant in the original dataset.

Usage

```r
## S3 method for class 'survFitTKTD'
plot(x, xlab = "Time", ylab = "Survival rate",
     main = NULL, concentration = NULL, spaghetti = FALSE,
     one.plot = FALSE, adddata = FALSE, addlegend = FALSE,
     style = "generic", ...)
```

Arguments

- `x` An object of class survFitTKTD.
- `xlab` A label for the X-axis, by default Time.
- `ylab` A label for the Y-axis, by default Survival rate.
- `main` A main title for the plot.
- `concentration` A numeric value corresponding to some concentration in data. If concentration = NULL, draws a plot for each concentration.
- `spaghetti` if TRUE, draws a set of survival curves using parameters drawn from the posterior distribution
- `one.plot` if TRUE, draws all the estimated curves in one plot instead of one per concentration.
- `adddata` if TRUE, adds the observed data to the plot with (frequentist) confidence intervals
- `addlegend` if TRUE, adds a default legend to the plot.
- `style` graphical backend, can be 'generic' or 'ggplot'
- `...` Further arguments to be passed to generic methods.

Details

The fitted curves represent the estimated survival rate as a function of time for each concentration. When adddata = TRUE the black dots depict the observed survival rate at each time point. Note that since our model does not take inter-replicate variability into consideration, replicates are systematically pooled in this plot. The function plots both 95% credible intervals for the estimated survival rate (by default the red area around the fitted curve) and 95% confidence intervals for the observed survival rate (as black error bars if adddata = TRUE). Both types of intervals are taken at the same level. Typically a good fit is expected to display a large overlap between the two intervals. If spaghetti = TRUE, the credible intervals are represented by two dotted lines limiting the credible band, and a spaghetti plot is added to this band. It consists of the representation of simulated curves using parameter values sampled in the posterior distribution (2% of the MCMC chains are randomly taken for this sample).
Examples

```r
# (1) Load the survival data
data(propiconazole)

# (2) Create an object of class "survData"
dat <- survData(propiconazole)

## Not run:
# (3) Run the survFitTKTD function
out <- survFitTKTD(dat)

# (4) Plot the fitted curves in one plot
plot(out)

# (5) Plot one fitted curve by concentration with credible limits as
# spaghetti, data and confidence intervals
# and with a ggplot style
plot(out, spaghetti = TRUE, adddata = TRUE, one.plot = FALSE,
     style = "ggplot")

# (6) Plot fitted curve for one specific concentration
plot(out, concentration = 36, style = "ggplot")

## End(Not run)
```

---

**plot.survFitTT**  
*Plotting method for survFitTT objects*

**Description**

This is the generic plot S3 method for the survFitTT class. It plots exposure-response fits for target time survival analysis.

**Usage**

```r
## S3 method for class 'survFitTT'
plot(x, xlab = "Concentration", ylab = "Survival rate",
     main = NULL, fitcol = "red", fitlty = 1, fitlw = 1,
     spaghetti = FALSE, cicol = "pink!", cilty = 1, cilwd = 1,
     adddata = FALSE, addlegend = FALSE, log.scale = FALSE,
     style = "generic", ...)
```

**Arguments**

- `x`  
  an object of class survFitTT

- `xlab`  
  a title for the x-axis
plot.survFitTT

ylab
main
fitcol
fitlty
fitlwd
spaghetti
cicol
cilty
cilwd
adddata
addlegend
log.scale
style
... Further arguments to be passed to generic methods

Details

The fitted curve represents the estimated survival rate after the target time has passed as a function of the concentration of pollutant; When adddata = TRUE the black dots depict the observed survival rate at each tested concentration. Note that since our model does not take inter-replicate variability into consideration, replicates are systematically pooled in this plot. The function plots both 95% credible intervals for the estimated survival rate (by default the red area around the fitted curve) and 95% confidence intervals for the observed survival rate (as black error bars if adddata = TRUE). Both types of intervals are taken at the same level. Typically a good fit is expected to display a large overlap between the two intervals. If spaghetti = TRUE, the credible intervals are represented by two dotted lines limiting the credible band, and a spaghetti plot is added to this band. It consists of the representation of simulated curves using parameter values sampled in the posterior distribution (10% of the MCMC chains are randomly taken for this sample).

Note

When style = "ggplot", the function calls function ggplot and returns an object of class ggplot.

Examples

# (1) Load the data
data(cadmium1)

# (2) Create an object of class "survData"
dat <- survData(cadmium1)

## Not run:
# (3) Run the survFitTT function with the log-logistic
#      binomial model
out <- survFitTT(dat, lx = c(5, 10, 15, 20, 30, 50, 80),
quiet = TRUE)

# (4) Plot the fitted curve
plot(out, log.scale = TRUE, adddata = TRUE)

# (5) Plot the fitted curve with ggplot style
plot(out, xlab = expression("Concentration in" ~ mu*g.L^(-1)),
     fitcol = "blue", adddata = TRUE, cicol = "blue",
     style = "ggplot")

## End(Not run)

---

**plotDoseResponse**

*Plot Dose-response from raw data*

**Description**

Plots the response as a function of the concentration (for a given target time).

**Usage**

```
plotDoseResponse(x, ...)```

**Arguments**

- `x`: an object used to select a method
- `...`: Further arguments to be passed to generic methods

---

**plotDoseResponse.reproData**

*Plotting method for reproData objects*

**Description**

This is the generic `plotDoseResponse` S3 method for the `reproData` class. It plots the number of offspring per individual-days as a function of concentration (for a given target time).

**Usage**

```
## S3 method for class 'reproData'
plotDoseResponse(x, xlab = "Concentration",
                 ylab = "Nb of offspring / Nb individual-days", main = NULL, ylim = NULL,
                 target.time = NULL, style = "generic", log.scale = FALSE,
                 remove.someLabels = FALSE, axis = TRUE, addlegend = TRUE, ...)```
Arguments

- `x`: an object of class `reproData`
- `xlab`: a title for the x-axis (optional)
- `ylab`: a label for the y-axis
- `main`: main title for the plot
- `ylim`: Y-axis limits
- `target.time`: a numeric value corresponding to some observed time in data
- `style`: graphical backend, can be 'generic' or 'ggplot'
- `log.scale`: if TRUE, displays x-axis in log scale
- `remove.someLabels`: if TRUE, removes 3/4 of X-axis labels in 'ggplot' style to avoid the label overlap
- `axis`: if TRUE displays ticks and label axis
- `add.legend`: if TRUE, adds a default legend to the plot
- ... Further arguments to be passed to generic methods

Details

The function plots the observed values of the reproduction rate (number of reproduction outputs per individual-day) for a given time as a function of concentration. The 95% Poisson confidence interval is added to each reproduction rate. It is calculated using function `pois.exact` from package `epitools`. As replicates are not pooled in this plot, overlapped points are shifted on the x-axis to help the visualization of replicates.

Note

When `style = "ggplot"`, the function calls function `ggplot` and returns an object of class `ggplot`.

See Also

- `pois.exact`

Examples

```r
library(ggplot2)

# (1) Load the data
data(zinc)
zinc <- reproData(zinc)

# (2) Plot dose-response
plotDoseResponse(zinc)

# (3) Plot dose-response with a ggplot style
plotDoseResponse(zinc, style = "ggplot")
```
plotDoseResponse.survData

Plotting method for survData objects

Description

This is the generic plotDoseResponse S3 method for the survData class. It plots the survival rate as a function of concentration (for a given target time).

Usage

```r
## S3 method for class 'survData'
plotDoseResponse(x, xlab = "Concentration", ylab = "Survival rate", main = NULL, target.time = NULL, style = "generic", log.scale = FALSE, remove.someLabels = FALSE, addlegend = TRUE, ...)
```

Arguments

- `x`: an object of class survData
- `xlab`: a title for the x-axis (optional)
- `ylab`: a label for the y-axis
- `main`: main title for the plot
- `target.time`: a numeric value corresponding to some observed time in data
- `style`: graphical backend, can be 'generic' or 'ggplot'
- `log.scale`: if TRUE, displays x-axis in log scale
- `remove.someLabels`: if TRUE, removes 3/4 of X-axis labels in 'ggplot' style to avoid the label overlap
- `addlegend`: if TRUE, adds a default legend to the plot
- `...`: Further arguments to be passed to generic methods

Details

The function plots the observed values of the survival rate for a given time as a function of concentration. The 95% binomial confidence interval is added to each survival rate. It is calculated using function `binom.test` from package stats. Replicates are systematically pooled in this plot.

Note

When `style = "ggplot"`, the function calls function `ggplot` and returns an object of class ggplot.

See Also

- `binom.test`
Examples

```r
library(ggplot2)

# (1) Load the data
data(zinc)
zinc <- survData(zinc)

# (2) Plot dose-response
plotDoseResponse(zinc)

# (3) Plot dose-response with a ggplot style
plotDoseResponse(zinc, style = "ggplot")
```

**Description**

Plots posterior predictive check for `reproFitTT`, `survFitTT` and `survFitTKTD` objects.

**Usage**

```r
ppc(x, ...)
```

**Arguments**

- `x` an object used to select a method
- `...` Further arguments to be passed to generic methods

**Description**

This is the generic ppc S3 method for the `reproFitTT` class. It plots the predicted values with 95% credible intervals versus the observed values.

**Usage**

```r
## S3 method for class 'reproFitTT'
ppc(x, style = "generic", ...)
```
Arguments

`x` An object of class `reproFitTT`
`style` graphical backend, can be 'generic' or 'ggplot'
`...` Further arguments to be passed to generic methods

Details

The coordinates of black points are the observed values of the cumulated number of reproduction outputs for a given concentration (x-scale) and the corresponding predicted values (y-scale). 95% prediction intervals are added to each predicted value, colored in green if this interval contains the observed value and in red in the other case. As replicates are not pooled in this plot, overlapped points are shifted on the x-axis to help the visualization of replicates. The bisecting line (y = x) is added to the plot in order to see if each prediction interval contains each observed value. As replicates are shifted on the x-axis, this line is represented by steps.

Examples

```r
# (1) Load the data
data(cadmium1)

# (2) Create an object of class "reproData"
dat <- reproData(cadmium1)

## Not run:
# (3) Run the reproFitTT function with the log-logistic gamma-poisson model
out <- reproFitTT(dat, stoc.part = "gammapoisson",
                  exc = c(5, 10, 15, 20, 30, 50, 80), quiet = TRUE)

# (4) Plot observed versus predicted values
ppc(out)

## End(Not run)
```

---

**Description**

This is the generic `ppc` S3 method for the `survFitTKTD` class. It plots the predicted values along with 95% credible intervals versus the observed values for `survFitTKTD` objects.

**Usage**

```
## S3 method for class 'survFitTKTD'
ppc(x, style = "generic", ...)
```
Arguments

- **x**: An object of class `survFitTKTD`
- **style**: Graphical backend, can be 'generic' or 'ggplot'
- **...**: Further arguments to be passed to generic methods

Details

The black points show the observed number of survivors (pooled replicates, on X-axis) against the corresponding predicted number (Y-axis). Predictions come along with 95% prediction intervals, which are depicted in green when they contain the observed value and in red otherwise. Samples with equal observed value are shifted on the X-axis. For that reason, the bisecting line (y = x), is represented by steps when observed values are low. That way we ensure green intervals do intersect the bisecting line.

Examples

```r
# (1) Load the data
data(propiconazole)

# (2) Create an object of class "survData"
dat <- survData(propiconazole)

## Not run:
# (3) Run the survFitTKTD function with the TKTD model
out <- survFitTKTD(dat)

# (4) Plot observed versus predicted values
ppc(out)

## End(Not run)
```

---

**ppc.survFitTT**  
*Posterior predictive check plot for survFitTT objects*

Description

This is the generic ppc S3 method for the survFitTT class. It plots the predicted values with 95% credible intervals versus the observed values for survFitTT objects.

Usage

```r
## S3 method for class 'survFitTT'
ppc(x, style = "generic", ...)
```
Arguments

- `x`: An object of class `survFitTT`
- `style`: Graphical package method: `generic` or `ggplot`
- `...`: Further arguments to be passed to generic methods

Details

The coordinates of black points are the observed values of the number of survivor (pooled replicates) for a given concentration (x-scale) and the corresponding predicted values (y-scale). 95 % prediction intervals are added to each predicted value, colored in green if this interval contains the observed value and in red in the other case. The bisecting line (y = x) is added to the plot in order to see if each prediction interval contains each observed value. As replicates are shifted on the x-axis, this line is represented by steps.

Examples

```
# (1) Load the data
data(cadmium1)

# (2) Create an object of class "survData"
dat <- survData(cadmium1)

## Not run:
# (3) Run the survFitTT function with the log-logistic binomial model
out <- survFitTT(dat, lcx = c(5, 10, 15, 20, 30, 50, 80),
quiet = TRUE)

# (4) Plot observed versus predicted values
ppc(out)

## End(Not run)
```

Description

This is the generic print S3 method for the `reproFitTT` class. It prints the underlying JAGS model and some information on the Bayesian inference procedure.

Usage

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

Arguments

x               An object of class reproFitTT

...            Further arguments to be passed to generic methods

Examples

# (1) Load the data
data(cadmium1)

# (2) Create a reproData object
cadmium1 <- reproData(cadmium1)

## Not run:
# (3) Run the reproFitTT function with the log-logistic
# model
out <- reproFitTT(cadmium1, ecx = c(5, 10, 15, 20, 30, 50, 80), quiet = TRUE)

# (4) Print the reproFitTT object
out

## End(Not run)

print.survFitTKTD  Print of survFitTKTD object

Description

This is the generic print S3 method for the survFitTKTD class. It prints the underlying JAGS model and some information on the Bayesian inference procedure.

Usage

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

Arguments

x               An object of class survFitTKTD

...            Further arguments to be passed to generic methods.
print.survFitTT

**Examples**

```r
# (1) Load the data
data(propiconazole)

# (2) Create a survData object
dat <- survData(propiconazole)

## Not run:
# (3) Run the survFitTKTD function
out <- survFitTT(dat, quiet = TRUE)

# (4) Print the survFitTKTD object
out

## End(Not run)
```

---

**print.survFitTT**  
*Print of survFitTT object*

**Description**

This is the generic print S3 method for the survFitTT class. It prints the underlying JAGS model and some information on the Bayesian inference procedure.

**Usage**

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

**Arguments**

- `x`  
  An object of class survFitTT

- `...`  
  Further arguments to be passed to generic methods

**Examples**

```r
# (1) Load the data
data(cadmium1)

# (2) Create a survData object
cadmium1 <- survData(cadmium1)

## Not run:
# (3) Run the survFitTT function with the log-logistic
# binomial model
out <- survFitTT(cadmium1, lcx = c(5, 10, 15, 20, 30, 50, 80), quiet = TRUE)
```
propiconazole

# (4) Print the survFitTT object out
## End(Not run)

<table>
<thead>
<tr>
<th>propiconazole</th>
<th>Survival datasets for Gammarus pulex exposed to propiconazole during 4 days</th>
</tr>
</thead>
</table>

### Description

Survival datasets of chronic laboratory bioassays with *Gammarus pulex* freshwater invertebrate exposed to eight concentrations of one fungicide (propiconazole) during 4 days. Height concentrations were tested with two replicates of 10 organisms per concentration. Survival is monitored at 5 time points.

### Usage

```r
data(propiconazole)
```

### Format

A data frame with 75 observations on the following four variables:

- **replicate**: A vector of class factor with the replicate code (SC for the control and A1 to code G2).
- **conc**: A vector of class numeric with propiconazole concentrations in mg.L^{-1}.
- **time**: A vector of class integer with the time points (in days from the beginning of the experiment $t = 0$).
- **Nsurv**: A vector of class integer with the number of alive individuals at each time point for each concentration and each replicate.

### References

reproData

*Creates a dataset for reproduction toxicity analysis*

**Description**

This function creates a `reproData` object from experimental data provided as a `data.frame`. The resulting object can then be used for plotting and model fitting. The `reproData` class is a sub-class of `survData`, meaning that all functions and method available for survival analysis can be used with `reproData` objects.

**Usage**

```r
reproData(x)
```

**Arguments**

- `x` a dataframe as expected by `survData` containing one additional `Nrepro` column of class `integer` with positive values only. This column should provide the number of offspring produced since the last observation.

**Details**

The `x` argument contains the experimental data, and should have the same structure that the argument of `survData`, plus a single additional column providing the total number of offspring observed since the last time point. The function fails if `x` does not meet the expected requirements. Please run `reproDataCheck` to ensure `x` is well-formed.

**Value**

An object of class `reproData`.

**Examples**

```r
# (1) Load reproduction dataset
data(cadmium1)

# (2) Create an object of class "reproData"
dat <- reproData(cadmium1)
class(dat)
```
reproDataCheck

Checks if an object can be used to perform reproduction toxicity data analysis

Description

The reproDataCheck function can be used to check if an object containing data from a reproduction toxicity assay meets the expectations of the function reproData.

Usage

reproDataCheck(data, diagnosis.plot = TRUE)

Arguments

data any object

diagnosis.plot if TRUE, may produce a diagnosis plot

Details

Since in MORSE reproduction datasets are a special case of survival datasets, reproDataCheck performs the same verifications than survDataCheck plus additional ones that are specific to reproduction data.

Value

The function returns a data.frame similar to the one returned by survDataCheck, except that it may contain the following additional error ids:

- NreproInteger: column Nrepro contains values of class other than integer
- Nrepro@T0: Nrepro is not 0 at time 0 for each concentration and each replicate
- Nsurv@Nrepro@T1P: at a giving time T, the number of alive individuals is null and the number of collected offspring is not null for the same replicate and the same concentration at time T + 1

Note

If an error of type dataframeExpected or missingColumn is detected, the function reproDataCheck is stopped. When no error is detected the reproDataCheck function returns an empty data.frame.

See Also

reproData
Examples

# Run the check data function
data(copper)
reproDataCheck(copper)

# Now we insert an error in the dataset, by setting a non-zero number of offspring at some time, although there is no surviving individual in the replicate from the previous time point.
copper$repro[148] <- as.integer(1)
reproDataCheck(copper)

---

reproFitTT

* Fits a Bayesian exposure-response model for target-time reproduction analysis *

Description

This function estimates the parameters of an exposure-response model for target-time reproduction analysis using Bayesian inference. In this model the response is the cumulated reproduction outputs of a population in a given time period in presence of mortality.

Usage

reproFitTT(data, stoc.part = "bestfit", target.time = NULL, ecx = c(5, 10, 20, 50), n.chains = 3, quiet = FALSE)

Arguments

data an object of class reproData
stoc.part stochastic part of the model. Possible values are "bestfit", "poisson" and "gammapoisson"
target.time defines the observation period. By default the last time point
ecx desired values of $x$ (in percent) for which to compute $EC_x$
n.chains number of MCMC chains. The minimum required number of chains is 2
quiet if TRUE, does not print messages and progress bars from JAGS

Details

Because some individuals may die during the observation period, the reproduction rate alone is not sufficient to account for the observed number of offspring. In addition, we need the time individuals have stayed alive during the experiment. The reproFitTT function estimates the number of individual-days in an experiment between its start and the target time. This covariable is then used to estimate a relation between the toxicant concentration and the reproduction rate per individual-day.
The `reprofittt` function fits two models, one where inter-individual variability is neglected ("Poisson" model) and one where it is taken into account ("gamma-Poisson" model). When setting `stoc.part` to "bestfit", a model comparison procedure is used to choose between them. More details are presented in the vignette accompanying the package.

**Value**

The function returns an object of class `reprofittt` which is a list of the following objects:

- **DIC**: DIC value of the selected model
- **estim.ECx**: a table of the estimated 5, 10, 20 and 50 % effective concentrations (by default) and their 95 % credible intervals
- **estim.par**: a table of the estimated parameters as medians and 95 % credible intervals
- **mcmc**: an object of class `mcmc.list` with the posterior distributions
- **model**: a JAGS model object
- **model.label**: a character string, "P" if the poisson model is used, "GP" if the gamma-poisson is used
- **parameters**: a list of the parameters names used in the model
- **n.chains**: an integer value corresponding to the number of chains used for the MCMC computation
- **n.iter**: a list of two indices indicating the beginning and the end of monitored iterations
- **n.thin**: a numerical value corresponding to the thinning interval
- **jags.data**: a list a the data passed to the jags model
- **transformed.data**: the `survData` object passed to the function
- **dataTT**: the dataset with which one the parameters are estimated

**Examples**

```r
# (1) Load the data
data(cadmium1)

# (2) Create an object of class "reproData"
dat <- reproData(cadmium1)

## Not run:
# (3) Run the reproFitTT function with the log-logistic gamma-poisson model
out <- reproFitTT(dat, stoc.part = "gammapiossom",
                  ecx = c(5, 10, 15, 20, 30, 50, 80), quiet = TRUE)

## End(Not run)
```
summary.reproData  

**Summary for reproData objects**

**Description**

This is the generic summary S3 method for the reproData class. It provides information about the structure of the dataset and the experimental design.

**Usage**

```r
## S3 method for class 'reproData'
summary(object, quiet = FALSE, ...)
```

**Arguments**

- `object`: an object of class reproData
- `quiet`: if TRUE, does no prints
- `...`: Further arguments to be passed to generic methods

**Value**

The function returns a list with the same fields than `summary.survData` plus an additional one:

- `NboffTimeConc`: nb of offspring for all concentrations and time points

**Examples**

```r
# (1) Load the data
data(cadmium1)

# (2) Create a reproData object
cadmium1 <- reproData(cadmium1)

# (3) Summarize the dataset
summary(cadmium1)
```

---

summary.reproFitTT  

**Summary for reproFitTT objects**

**Description**

This is the generic summary S3 method for the reproFitTT class. It shows the quantiles of priors and posteriors on parameters and the quantiles of the posterior on the ECx.
Usage

```r
## S3 method for class 'reproFitTT'
summary(object, quiet = FALSE, ...)
```

Arguments

- `object`: an object of class `reproFitTT`
- `quiet`: when `FALSE`, prints summary on standard output
- `...`: Further arguments to be passed to generic methods

Value

The function returns a list with the following fields:

- `qpriors`: quantiles for the model's prior
- `qposteriors`: quantiles for the model's posteriors
- `qECX`: quantiles for ECX values

Examples

```r
# (1) Load the data
data(cadmium1)

data(cadmium1)  

# (2) Create a reproData object
cadmium1 <- reproData(cadmium1)

cadmium1 <- reproData(cadmium1)

## Not run:
# (3) Run the reproFitTT function with the log-logistic
# model
out <- reproFitTT(cadmium1, ecx = c(5, 10, 15, 20, 30, 50, 80),
quiet = TRUE)

# (4) summarize the reproFitTT object
summary(out)

## End(Not run)
```

Description

The generic `summary` S3 method for the `survData` class provides information about the structure of the dataset and the experimental design.
Usage

```r
## S3 method for class 'survData'
summary(object, quiet = FALSE, ...)
```

Arguments

- `object`: an object of class `survData`
- `quiet`: when `FALSE`, prints summary on standard output
- `...`: Further arguments to be passed to generic methods

Value

The function returns a list with the following fields:

- `NbrepTimeConc`: nb of replicates for all concentrations and time points
- `NbsurvTimeConc`: nb of surviving ind. for all concentrations and time points

Examples

```r
# (1) Load the data
data(cadmium1)

# (2) Create a survData object
cadmium1 <- survData(cadmium1)

# (3) Summarize the dataset
summary(cadmium1)
```

---

**summary.survFitTKTD**

Summary for `survFitTKTD` objects

Description

This is the generic `summary` S3 method for the `survFitTKTD` class. It shows the quantiles of priors and posteriors on parameters.

Usage

```r
## S3 method for class 'survFitTKTD'
summary(object, quiet = FALSE, ...)
```

Arguments

- `object`: an object of class `survFitTKTD`
- `quiet`: when `FALSE`, prints summary on standard output
- `...`: Further arguments to be passed to generic methods.
summary.survFitTT

Value

The function returns a list with the following fields:

- Qpriors  quantiles for the model’s prior
- Qposteriors  quantiles for the model’s posteriors

Examples

# (1) Load the data
data(propiconazole)

# (2) Create a survData object
dat <- survData(propiconazole)

## Not run:
# (3) Run the survFitTKTD function
out <- survFitTKTD(dat)

# (4) Summarize the survFitTKTD object
summary(out)

## End(Not run)

summary.survFitTT  Summary for survFitTT objects

Description

This is the generic summary S3 method for the survFitTT class. It shows the quantiles of priors and posteriors on parameters and the quantiles of the posterior on the LCx.

Usage

## S3 method for class 'survFitTT'
summary(object, quiet = FALSE, ...)

Arguments

- object  an object of class survFitTT
- quiet  when FALSE, prints summary on standard output
- ...  Further arguments to be passed to generic methods

Value

The function returns a list with the following fields:

- Qpriors  quantiles for the model’s prior
- Qposteriors  quantiles for the model’s posteriors
-QLCx  quantiles for LCx values
survData

Creating a dataset for survival analysis

Description

This function creates a survData object from experimental data provided as a data.frame. The resulting object can then be used for plotting and model fitting. It can also be used to generate individual-time estimates.

Usage

survData(data)

Arguments

data a data.frame containing the following four columns:
  • replicate: a vector of class integer or factor for replicate identification
  • conc: a vector of class numeric with tested concentrations (positive values)
  • time: a vector of class integer with time points, min value must be 0
  • Nsurv: a vector of class integer providing the number of alive individuals at each time point for each concentration and each replicate

Details

The data argument describes experimental results from a survival assay. Each line of the data.frame corresponds to one experimental measurement, that is a number of alive individuals for a given concentration of pollutant at a certain time during the assay in a certain replicate. The function fails if data does not meet the expected requirements. Please run survDataCheck to ensure data is well-formed.
Value

A dataframe of class survData.

See Also

survDataCheck

Examples

# (1) Load the survival dataset
data(zinc)

# (2) Create an objet of class 'survData'
dat <- survData(zinc)
class(dat)

```
# survDataCheck
Checks if an object can be used to perform survival analysis
```

Description

The survDataCheck function can be used to check if an object containing survival data is formatted according to the expectations of the survData function.

Usage

survDataCheck(data, diagnosis.plot = TRUE)

Arguments

data any object
diagnosis.plot if TRUE, the function may produce diagnosis plots

Value

The function returns a dataframe with two columns id and msg of character strings. When no error is detected this dataframe is empty. Here is the list of possible error ids and their signification:

dataframeExpected an object of class data.frame is expected
missingColumn at least one expected column heading is missing
firstTime0 the first time point for some (concentration, replicate) is not 0
concNumeric column conc contains a value of class other than numeric
timeNumeric column time contains a value of class other than numeric
NsurvInteger column Nsurv contains a value of class other than integer
tablePositive some data are negative
Nsurv0T0 Nsurv is 0 at time 0 for some (concentration, replicate)
survFitTKTD

Fits a TKTD for survival analysis using Bayesian inference

Description

This function estimates the parameters of a TKTD model for survival analysis using Bayesian inference. In this model, the survival rate of individuals is modeled as a function of the pollutant’s concentration with a mechanistic description of toxic effects on survival over time.

Usage

survFitTKTD(data, n.chains = 3, quiet = FALSE)

Arguments

data  
An object of class survData.

n.chains  
Number of MCMC chains. The minimum required number of chains is 2.

quiet  
If FALSE, prints logs and progress bar from JAGS.
Details

Details of the model are presented in the vignette accompanying the package.

Value

The function returns an object of class `survfitTKTD`, which is a list with the following fields:

- `estim.par`: a table of the estimated parameters (medians) and 95% credible intervals
- `mcmc`: an object of class `mcmc.list` with the posterior distributions
- `model`: a JAGS model object
- `parameters`: a list of the parameters names used in the model
- `n.chains`: an integer value corresponding to the number of chains used for the MCMC computation
- `n.iter`: a list of two indices indicating the beginning and end of monitored iterations
- `n.thin`: a numerical value corresponding to the thinning interval
- `jags.data`: a list a the data passed to the jags model

Examples

```r
# (1) Load the survival data
data(propiconazole)

# (2) Create an object of class "survData"
dat <- survData(propiconazole)

## Not run:
# (3) Run the survFitTKTD function
out <- survFitTKTD(dat)

# (4) Summary look the estimated values (parameters)
summary(out)

# (5) Plot the fitted curve
plot(out, adddata = TRUE)

# (6) Plot the fitted curve with ggplot style and CI as spaghetti
plot(out, spaghetti = TRUE, adddata = TRUE,
     style = "ggplot")

## End(Not run)
```
survFitTT **Fits a Bayesian exposure-response model for target-time survival analysis**

Description

This function estimates the parameters of an exposure-response model for target-time survival analysis using Bayesian inference. In this model, the survival rate of individuals after some time (called target time) is modeled as a function of the pollutant’s concentration. The actual number of surviving individuals is then modeled as a stochastic function of the survival rate. Details of the model are presented in the vignette accompanying the package.

Usage

`survFitTT(data, target.time = NULL, lcx = c(5, 10, 20, 50), n.chains = 3, quiet = FALSE)`

Arguments

- `data`: an object of class `survData`
- `target.time`: the chosen endpoint to evaluate the effect of a given concentration of pollutant, by default the last time point available for all concentrations
- `lcx`: desired values of \( x \) (in percent) for which to compute \( LC_x \)
- `n.chains`: number of MCMC chains, the minimum required number of chains is 2
- `quiet`: if `TRUE`, does not print messages and progress bars from JAGS

Details

The function returns parameter estimates of the exposure-response model and estimates of the so-called \( LC_x \), that is the concentration of pollutant required to obtain a 1 - \( x \) survival rate.

Value

The function returns an object of class `survFitTT`, which is a list with the following fields:

- `estim.Lcx`: a table of the estimated \( LC_x \) along with their 95% credible intervals
- `estim.par`: a table of the estimated parameters (medians) and 95% credible intervals
- `det.part`: the name of the deterministic part of the used model
- `mcmc`: an object of class `mcmc.list` with the posterior distributions
- `model`: a JAGS model object
- `parameters`: a list of the parameters names used in the model
- `n.chains`: an integer value corresponding to the number of chains used for the MCMC computation
- `n.iter`: a list of two indices indicating the beginning and end of monitored iterations
n.thin a numerical value corresponding to the thinning interval
jags.data a list of the data passed to the jags model
transformed.data the survData object passed to the function
dataTT the dataset with which one the parameters are estimated

Examples

# (1) Load the data
data(cadmium1)

# (2) Create an object of class "survData"
dat <- survData(cadmium1)

## Not run:
## (3) Run the survFitTT function with the log-logistic
## binomial model
out <- survFitTT(dat, lcx = c(5, 10, 15, 20, 30, 50, 80),
                 quiet = TRUE)

## End(Not run)

zinc Reproduction and survival datasets for Daphnia magna exposed to zinc during 21 days

Description

Reproduction and survival datasets of chronic laboratory bioassays with Daphnia magna freshwater invertebrate exposed to four concentrations of one metal contaminant (zinc) during 21 days. Four concentrations were tested with three replicates per concentration. Each replicate contained 20 organisms. Reproduction and survival were monitored at 15 time points.

Usage

data(zinc)

Format

A data frame with 180 observations on the following five variables:

replicate A vector of class factor with the replicate code (A to C).
conc A vector of class numeric with zinc concentrations in mg.L$^{-1}$.
time A vector of class integer with the time points (in days from the beginning of the experiment $t = 0$).
\texttt{Nsurv} A vector of class \texttt{integer} with the number of alive individuals at each time point for each concentration and each replicate.

\texttt{Nrepro} A vector of class \texttt{integer} with the number of offspring at each time point for each concentration and each replicate.

References

Index

*Topic check
reproDataCheck, 27
*Topic datasets
cadmium1, 5
cadmium2, 5
chlordan, 6
copper, 7
dichromate, 8
propiconazole, 25
zinc, 39
*Topic estimation
reprofitt, 28
survfittktd, 36
survfitt, 38
*Topic package
morse-package, 3
*Topic plot
plot.reproData, 9
plot.reprofitt, 10
plot.survData, 11
plot.survFitTKTD, 13
plot.survFitTT, 14
plotDoseResponse.reproData, 16
plotDoseResponse.survData, 18
*Topic print
print.reprofitt, 22
print.survFitTKTD, 23
print.survFitTT, 24
*Topic summary
summary.reproData, 30
summary.reprofitt, 30
summary.survData, 31
summary.survFitTKTD, 32
summary.survFitTT, 33
*Topic transformation
reproData, 26
survData, 34
as.survData, 4

binom.test, 18
cadmium1, 5
cadmium2, 5
chlordan, 6
copper, 7
dichromate, 8
ggplot, 4, 9, 11, 12, 15, 17, 18
morse (morse-package), 3
morse-package, 3
plot.reproData, 9
plot.reprofitt, 10
plot.survData, 11
plot.survFitTKTD, 13
plot.survFitTT, 14
plotDoseResponse, 16
plotDoseResponse.reproData, 16
plotDoseResponse.survData, 18
pois.exact, 17
ppc, 19
ppc.reprofitt, 19
ppc.survFitTKTD, 20
ppc.survFitTT, 21
print.reprofitt, 22
print.survFitTKTD, 23
print.survFitTT, 24
propiconazole, 25
reproData, 4, 26, 27
reproDataCheck, 26, 27
reprofitt, 28
rjags, 4

summary.reproData, 30
summary.reprofitt, 30
summary.survData, 30, 31
summary.survFitTKTD, 32

41
summary.survFitTT, 33
survData, 4, 34, 36
survDataCheck, 27, 34, 35, 35
survFitTKTD, 36
survFitTT, 38

zinc, 39