Package 'dalmatian'

January 29, 2018

Title Automating the Fitting of Double Linear Mixed Models in 'JAGS'

Version 0.3.0

Date 2018-01-19

Description Automates fitting of double GLM in 'JAGS'. Includes automatic generation of 'JAGS' scripts, running 'JAGS' via 'rjags', and summarizing the resulting output.

Depends R (>= 3.0.0)

License GPL-2

LazyData true

VignetteBuilder knitr

Imports rjags, coda, gdata, ggmcmc, dglm

Suggests knitr, ggplot2, dplyr

RoxygenNote 6.0.1

NeedsCompilation no

Author Simon Bonner [aut, cre], Hanna Kim [aut]

Maintainer Simon Bonner <sbonner6@uwo.ca>

Repository CRAN

Date/Publication 2018-01-29 12:44:05 UTC

R topics documented:

`caterpillar` *Caterpillar (Generic)*

Description

Caterpillar (Generic)

Usage

```
caterpillar(object, ...)
```

Arguments

`object`	Object to assess.
`...`	Ignored

Examples

```
## Load output from previously run model
load(system.file("Pied_Flycatchers_1","pfresults.RData",package="dalmatian"))

## Generate caterpillar
pfcaterpillar <- caterpillar(pfresults,plot = FALSE)
```

`caterpillar.dalmatian` *Caterpillar (dalmatian)*

Description

Construct caterpillar plots for key (or selected) parameters in a dalmatian object.

Usage

```
## S3 method for class 'dalmatian'
caterpillar(object, family = NULL,
  nstart = start(object$coda), nend = end(object$coda),
  nthin = thin(object$coda), plot = TRUE, ...)
```

Arguments

object	Object of class dalmatian created by dalmatian().
family	String defining selected family of variables (see help for ggs()).
nstart	Start point for computing summary statistics (relative to true start of chain).
nend	End point for computing summary statistics (relative to true start of chain).
nthin	Thinning factor for computing summary statsitics (relative to full chain and not previously thinned output).
plot	If TRUE then generate plots. Otherwise, a list of ggplot objects will be returned.
...	Ignored

Value

A list of ggplot objects that can be used to later reproduce the plots via print.

Examples

```
## Load output from previously run model
load(system.file("Pied_Flycatchers_1","pfresults.RData",package="dalmatian"))

## Generate caterpillar
pfcaterpillar <- caterpillar(pfresults,plot = FALSE)
```

---

convergence *Convergence Diagnostics (S3 Generic)*

Description

Generic function for computing convergence diagnostics.

Usage

```
convergence(object, ...)
```

Arguments

object	Object to asses.
...	Ignored

Examples

```
## Load output from previously run model
load(system.file("Pied_Flycatchers_1","pfresults.RData",package="dalmatian"))

## Compute convergence diagnostics
pfconvergence <- convergence(pfresults)
```

`convergence.dalmatian` *Convergence*

Description

Compute convergence diagnostics for a dalmatian object.

Usage

```
## S3 method for class 'dalmatian'
convergence(object, pars = NULL,
  nstart = start(object$coda), nend = end(object$coda),
  nthin = coda::thin(object$coda), raftery = NULL, ...)
```

Arguments

`object`	Object of class `dalmatian` created by `dalmatian()`.
`pars`	List of parameters to assess. If NULL (default) then diagnostics are computed for the fixed effects and random effects standard deviations in both the mean and variance models.
`nstart`	Start point for computing summary statistics (relative to true start of chain).
`nend`	End point for computing summary statistics (relative to true start of chain).
`nthin`	Thinning factor for computing summary statsitics (relative to full chain and not previously thinned output).
`raftery`	List of arguments to be passed to `raftery.diag()`. Any values not provided will be set to their defaults (see `help(raftery.diag())` for details).
`...`	Ignored

Value

List containing Gelman-Rubin and Raftery convergence diagnostics and effective sampel sizes for the selected parameters.

Examples

```
## Load output from previously run model
load(system.file("Pied_Flycatchers_1","pfresults.RData",package="dalmatian"))

## Compute convergence diagnostics
pfconvergence <- convergence(pfresults)
```

---

dalmatian *Run DGLM in* JAGS *via* rjags

Description

The primary function which automates the running of JAGS.

Usage

```
dalmatian(df, mean.model, variance.model, jags.model.args, coda.samples.args,
  response = NULL, rounding = FALSE, lower = NULL, upper = NULL,
  parameters = NULL, svd = TRUE, residuals = FALSE, gencode = NULL,
  drop.levels = TRUE, drop.missing = TRUE, overwrite = FALSE,
  debug = FALSE, saveJAGSinput = NULL)
```

Arguments

df	Data frame containing the response and predictor values for each individual. (data.frame)
mean.model	Model list specifying the structure of the mean. (list)
variance.model	Model list specifying the structure of the variance. (list)
jags.model.args	List containing named arguments of jags.model. (list)
coda.samples.args	List containing named arguments of coda.samples. (list)
response	Name of variable in the data frame representing the response. (character)
rounding	Specifies that response has been rounded if TRUE. (logical)
lower	Name of variable in the data frame representing the lower bound on the response if rounded. (character)
upper	Name of variable in the data frame representing the upper bound on the response if rounded. (character)
parameters	Names of parameters to monitor. If NULL then default values are selected. (character)
svd	Compute Singular Variable Decomposition of model matrices to improve convergence. (logical)

residuals	If TRUE then compute residuals in output. (logical)
gencode	If TRUE then generate code potentially overwriting existing model file. By default generate code if the file does not exist and prompt user if it does. (logical)
drop.levels	If TRUE then drop unused levels from all factors in df. (logical)
drop.missing	If TRUE then remove records with missing response variable. (logical)
overwrite	If TRUE then overwrite existing JAGS files (non-interactive sessions only). (logical)
debug	If TRUE then enter debug model. (logical)
saveJAGSinput	Directory to which jags.model input is saved prior to calling jags.model(). This is useful for debugging. No files saved if NULL. (character)

Details

The primary function in the package, dalmatian automates the generation of code, data, and initial values. These are then passed as arguments to function from the rjags package which automates the generation of samplse from the posterior.

Value

samples (mcmc.list)

Author(s)

Simon Bonner

Examples

```
## Not run:
## Load pied flycatcher data
data(pied_flycatchers_1)

## Create variables bounding the true load
pfdata$lower=ifelse(pfdata$load==0,log(.001),log(pfdata$load-.049))
pfdata$upper=log(pfdata$load+.05)
## Mean model
mymean=list(fixed=list(name="alpha",
                       formula=~ log(IVI) + broodsize + sex,
                       priors=list(c("dnorm",0,.001))))

## Variance model
myvar=list(fixed=list(name="psi",
                       link="log",
                       formula=~broodsize + sex,
                       priors=list(c("dnorm",0,.001))))

## Set working directory
## By default uses a system temp directory. You probably want to change this.
workingDir <- tempdir()
```

```
## Define list of arguments for jags.model()
jm.args <- list(file=file.path(workingDir,"pied_flycatcher_1_jags.R"),n.adapt=1000)

## Define list of arguments for coda.samples()
cs.args <- list(n.iter=5000)

## Run the model using dalmatian
pfresults <- dalmatian(df=pfdata,
                       mean.model=mymean,
                       variance.model=myvar,
                       jags.model.args=jm.args,
                       coda.samples.args=cs.args,
                       rounding=TRUE,
                       lower="lower",
                       upper="upper",
                       debug=FALSE)

## End(Not run)
```

`fitted.dalmatian` *Prediction method for dalmatian Fitted Objects*

Description

Prediction method for dalmatian Fitted Objects

Usage

```
## S3 method for class 'dalmatian'
fitted(object, df = object$df, method = "mean",
  ci = TRUE, level = 0.95, ...)
```

Arguments

`object`	Object of class `dalmatian` created by `dalmatian()`.
`df`	data frame containing predictor values to predict response variables. Defaults to data in `object` if not supplied. (data.frame)
`method`	Method to construct the fitted model. Either `"mean"` or `"mode"` (character)
`ci`	returning credible intervals for predictions if TRUE (logical)
`level`	level of credible intervals for predictions (numeric)
`...`	Ignored

Value

predictions (list)

Examples

```
## Load pied flycatcher data
data(pied_flycatchers_1)

## Create variables bounding the true load
pfdata$lower=ifelse(pfdata$load==0,log(.001),log(pfdata$load-.049))
pfdata$upper=log(pfdata$load+.05)

## Add 'log(IVI)' variable in pfdata
pfdata$'log(IVI)' <- log(pfdata$IVI)

## Load output from previously run model
load(system.file("Pied_Flycatchers_1","pfresults.RData",package="dalmatian"))

## Compute fitted values
pred.pfresults <- fitted(object = pfresults,
                         df = pfdata,
                         method = "mean",
                         ci = TRUE,
                         level = 0.95)
```

---

pfdata *Pied flycatcher feeding data*

Description

Dataset containing 5795 records of 60 pied flycatchers from 33 nest boxes feeding their nestlings during a brood manipulation experiment.

Usage

```
pfdata
```

Format

A data frame containing 5795 rows and 17 variables

---

print.dalmatian.summary *Print Summary (dalmatian)*

Description

Print Summary (dalmatian)

Usage

```
## S3 method for class 'dalmatian.summary'
print(x, digits = 2, ...)
```

Arguments

x	Object of class dalamtion.summary created by summary.dalmatian().
digits	Number of digits to display after decimal.
...	Ignored

Examples

```
## Load output from previously run model
load(system.file("Pied_Flycatchers_1","pfresults.RData",package="dalmatian"))

## Compute numerical summaries
print(summary(pfresults))
```

---

ranef *Random Effects (S3 Generic)*

Description

Generic function for exporting summaries of random effects.

Usage

```
ranef(object, ...)
```

Arguments

object	Input object
...	Ignored

Examples

```
## Load output from previously run model
load(system.file("Pied_Flycatchers_1","pfresults.RData",package="dalmatian"))

## Compute numerical summaries
ranef(pfresults)
```

ranef.dalmatian *Random Effects (dalmatian)*

Description

Compute posterior summary statistics for the individual random effects in each part of the model.

Usage

```
## S3 method for class 'dalmatian'
ranef(object, nstart = start(object$coda),
  nend = end(object$coda), nthin = thin(object$coda), ...)
```

Arguments

object	Object of class dalmatian created by dalmatian().
nstart	Start point for computing summary statistics (relative to true start of chain).
nend	End point for computing summary statistics (relative to true start of chain).
nthin	Thinning factor for computing summary statsitics (relative to full chain and not previously thinned output).
...	Ignored

Value

output (list)

Examples

```
## Load output from previously run model
load(system.file("Pied_Flycatchers_1","pfresults.RData",package="dalmatian"))

## Compute numerical summaries
ranef(pfresults)
```

setJAGSInits *Set initial values for* dalmatian

Description

Set initial values for dalmatian

Usage

```
setJAGSInits(mean.model, variance.model, fixed.mean = NULL,
  fixed.variance = NULL, y = NULL, random.mean = NULL, sd.mean = NULL,
  random.variance = NULL, sd.variance = NULL)
```

Arguments

mean.model	Model list specifying the structure of the mean. (list)
variance.model	Model list specifyint the structure of the variance. (list)
fixed.mean	Initial values for the fixed effects of the mean. (numeric)
fixed.variance	Initial values for the fixed effects of the variance. (numeric)
y	Inital values for the true response. This should only be specified if the rounding = TRUE in the main call to dalmatian.
random.mean	Initial values for the random effects of the mean. (numeric)
sd.mean	Initial values for the standard deviation of the random effects of the mean. (numeric)
random.variance	Initial values for the random effects of the variance. (numeric
sd.variance	Initial values for the standard deviation of the random effects of the variance. (numeric)

Details

Allows the user to set initial values for dalmatian. Any values not specified will by initialized by JAGS.

Value

inits (list)

Author(s)

Simon Bonner

Examples

```
## Load pied flycatcher data
data(pied_flycatchers_1)

## Create variables bounding the true load
pfdata$lower=ifelse(pfdata$load==0,log(.001),log(pfdata$load-.049))
pfdata$upper=log(pfdata$load+.05)

## Load output from previously run model
load(system.file("Pied_Flycatchers_1","pfresults.RData",package="dalmatian"))

## Set initial values for a new run of the same model
inits <- lapply(1:3,function(i){
```

```
  setJAGSInits(pfresults$mean.model,
               pfresults$variance.model,
               y = runif(nrow(pfdata),pfdata$lower,pfdata$upper),
               fixed.mean = tail(pfresults$coda[[i]],1)[1:4],
               fixed.variance = tail(pfresults$coda[[i]],1)[5:7],
               sd.mean = 1)
})
```

`summary.dalmatian` *Summary (dalmatian)*

Description

Summary (dalmatian)

Usage

```
## S3 method for class 'dalmatian'
summary(object, nstart = start(object$coda),
  nend = end(object$coda), nthin = thin(object$coda), ...)
```

Arguments

`object`	Object of class `dalmatian` created by `dalmatian()`.
`nstart`	Start point for computing summary statistics (relative to true start of chain).
`nend`	End point for computing summary statistics (relative to true start of chain).
`nthin`	Thinning factor for computing summary statsitics (relative to full chain and not previously thinned output).
`...`	Ignored

Value

output (list)

Examples

```
## Load output from previously run model
load(system.file("Pied_Flycatchers_1","pfresults.RData",package="dalmatian"))

## Compute numerical summaries
summary(pfresults)
```

traceplots *Traceplots (Generic)*

Description

Traceplots (Generic)

Usage

```
traceplots(object, ...)
```

Arguments

object	Object to assess.
...	Ignored

Examples

```
## Load output from previously run model
load(system.file("Pied_Flycatchers_1","pfresults.RData",package="dalmatian"))

## Generate traceplots
pftraceplots <- traceplots(pfresults)
```

traceplots.dalmatian *Traceplots (dalmatian)*

Description

Construct traceplots for key (or selected) parameters in a dalmatian object.

Usage

```
## S3 method for class 'dalmatian'
traceplots(object, family = NULL,
  nstart = start(object$coda), nend = end(object$coda),
  nthin = thin(object$coda), plot = TRUE, ...)
```

Arguments

object	Object of class dalmatian created by dalmatian().
family	String defining selected family of variables (see help for ggs()).
nstart	Start point for computing summary statistics (relative to true start of chain).
nend	End point for computing summary statistics (relative to true start of chain).
nthin	Thinning factor for computing summary statsitics (relative to full chain and not previously thinned output).
plot	If TRUE then generate plots. Otherwise, a list of ggplot objects will be returned.
...	Ignored

Value

A list of ggplot objects that can be used to later reproduce the plots via print.

Examples

```
## Load output from previously run model
load(system.file("Pied_Flycatchers_1","pfresults.RData",package="dalmatian"))

## Generate traceplots
pftraceplots <- traceplots(pfresults)
```

---

weights.df *Simulated data for illustrating the use of weights*

Description

Simulated data for illustrating the use of weights in the particular case when the responses are averages of observed with different denominators

Usage

```
weights.df
```

Format

An object of class data.frame with 100 rows and 3 columns.

Details

@format A data frame with 100 rows and 3 columns:

n The number of observations.

x The common predictor value.

y The mean response value.

Index