Package ‘eggCounts’

July 22, 2025

Imports methods, utils, rstan (>= 2.26), rstantools (>= 2.3.1), boot, coda, numbers, lattice, rootSolve

Depends R (>= 3.4.0), Rcpp (>= 0.12.0)

Suggests eggCountsExtra, R.rsp, testthat

VignetteBuilder R.rsp

Title Hierarchical Modelling of Faecal Egg Counts

Version 2.4

Date 2023-10-14

Maintainer Craig Wang <craigwang247@gmail.com>

Description An implementation of Bayesian hierarchical models for faecal egg count data to assess anthelmintic efficacy. Bayesian inference is done via MCMC sampling using 'Stan' <https://mc-stan.org/>.

SystemRequirements GNU make

Additional_repositories https://craigwangstat.github.io/eggCountsExtra-package/

License GPL (>= 3)

LinkingTo rstan (>= 2.26), Rcpp (>= 1.0.7), BH (>= 1.75.0), StanHeaders (>= 2.26.0), RcppEigen (>= 0.3.3.9.1), RcppParallel (>= 5.1.4)

LazyLoad yes

LazyData true

ByteCompile true

Biarch true

NeedsCompilation yes

URL https://www.math.uzh.ch/pages/eggcount/

RcppModules stan_fit4paired_mod, stan_fit4unpaired_mod, stan_fit4zipaired_mod, stan_fit4ziunpaired_mod, stan_fit4nb_mod, stan_fit4zinb_mod, stan_fit4indefficacy_mod, stan_fit4simple_mod

Author Craig Wang [aut, cre] (ORCID: <https://orcid.org/0000-0003-1804-2463>),
Michaela Paul [aut],
Tea Isler [ctb],
Reinhard Furrer [ctb] (ORCID: <https://orcid.org/0000-0002-6319-2332>),
Trustees of Columbia University [cph] (src/init.cpp, tools/make_cc.R, R/stanmodels.R)

Repository CRAN

Date/Publication 2023-10-14 22:22:34 UTC

Contents

`eggCounts-package` *Hierarchical modelling of faecal egg counts*

Description

This package implements Bayesian hierarchical models for the analysis of faecal egg count data. Bayesian inference is done via efficient MCMC sampling using Stan. Additional (experimental) models are available externally for handling FECs with potential outliers or bi-modality. The models are in **eggCountsExtra** package hosted on Github.

Details

Package:	eggCounts
Type:	Package
Version:	2.4
Date:	2023-10-14
License:	GPL (>= 3)
LazyLoad:	yes

About Stan

Stan is a probabilistic programming language for specifying Bayesian hierarchical models. It is computationally faster compared to conventional MCMC techniques. For the installation instruction and other information about Stan, please read here.

Author(s)

Craig Wang `<craig.wang@math.uzh.ch>`
Michaela Paul

Examples

```
## Not run:

## Citations
citation('eggCounts')

## History of changes
file.show(system.file("NEWS", package = "eggCounts"))

## Demonstration
demo("fecm_stan", package = "eggCounts")

## Install eggCountsExtra
devtools::install_github("CraigWangUZH/eggCountsExtra")

## End(Not run)
```

---

epgs — *Faecal egg count samples (before and after treatment)*

Description

This is an example dataset containing 14 eggs per gram (epg) values in sheep before and after anthelmintic treatment of benzimidazole. The correction factor of the diagnostic technique was 50.

Usage

```
data(epgs)
```

Format

A data.frame containing 14 observations.

References

Craig Wang, Paul R. Torgerson, Johan Hoglund, Reinhard Furrer, Zero-inflated hierarchical models for faecal egg counts to assess anthelmintic efficacy, Veterinary Parasitology, Volume 235, 2017, Pages 20-28.

---

fecrtCI — *Compute standard FECRT according to WAAVP guidelines*

Description

Computes the standard Faecal Egg Count Reduction Test together with approximate confidence interval according to the WAAVP guidelines (Coles et al., 1992, 2006). The function also returns bootstrap confidence intervals.

Usage

```
fecrtCI(epg1, epg2, paired = FALSE, alpha = 0.05, R = 1999)
```

Arguments

`epg1`	numeric vector. Faecal egg counts in untreated animals.
`epg2`	numeric vector. Faecal egg counts in treated animals.
`paired`	logical. If TRUE, indicates samples are paired. Otherwise samples are unpaired.
`alpha`	numeric. Confidence level of the interval.
`R`	numeric. Number of bootstrap replicates.

Value

A list with

`estimate`	estimated percentage reduction in mean epg
`bootCI`	bootstrap confidence interval
`approxCI`	approximate confidence interval

Author(s)

Michaela Paul

References

Coles GC, Bauer C, Borgsteede FHM, Geerts S, Klei TR, Taylor MA and Waller, PJ (1992). World Association for the Advancement of Veterinary Parasitology (WAAVP) methods for the detection of anthelmintic resistance in nematodes of veterinary importance, Veterinary Parasitology, 44:35-44.

Coles GC, Jackson F, Pomroy WE, Prichard RK, von Samson-Himmelstjerna G, Silvestre A, Taylor MA and Vercruysse J (2006). The detection of anthelmintic resistance in nematodes of veterinary importance, Veterinary Parasitology, 136:167-185.

Examples

```
data(epgs)
fecrtCI(epgs$before, epgs$after, paired = TRUE)
```

---

fecr_probs *Compute probability of the reduction parameter relative to a given threshold*

Description

Computes probability of the reduction parameter's marginal posterior density relative to a threshold.

Usage

```
fecr_probs(stanFit, threshold = 0.95, lessthan = TRUE,
           plot = TRUE, xlab, ylab, main, verbose = TRUE, ...)
```

Arguments

`stanFit`	a `stanfit` object from the output of `fecr_stan()`.
`threshold`	numeric. The default threshold is 0.95 (95%).
`lessthan`	logical. If TRUE, the probability less than the threshold is computed. Otherwise greater or equal to the threshold is computed. Default is TRUE.
`plot`	logical. If TRUE, the posterior density of the reduction is plotted with region less than the threshold shaded.
`xlab, ylab, main`	strings. Arguments for plotting. Only used if `plot = TRUE`.
`verbose`	logical. If TRUE, a statement with computed probability is printed.
`...`	additional plotting arguments

Value

Returns a numeric value indicating the probability in percentage.

Author(s)

Craig Wang

Examples

```
## load sample data
data(epgs)

## apply individual efficacy model to the data vectors
model <- fecr_stan(epgs$before, epgs$after, rawCounts = FALSE, preCF = 10,
                   paired = TRUE, indEfficacy = TRUE)
fecr_probs(model$stan.samples)
```

`fecr_stan` *Model the reduction of faecal egg count*

Description

Models the reduction in faecal egg counts with Bayesian hierarchical models. See Details for a list of model choices.

Usage

```
fecr_stan(preFEC, postFEC, rawCounts = FALSE, preCF = 50, postCF = preCF,
  paired = TRUE, indEfficacy = TRUE, zeroInflation = FALSE,
  muPrior, kappaPrior, deltaPrior, phiPrior, deltakappaPrior,
  nsamples = 2000, nburnin = 1000, thinning = 1, nchain = 2,
  ncore = 1, adaptDelta = 0.95, saveAll = FALSE, verbose = FALSE)
```

Arguments

`preFEC`	numeric vector. Pre-treatment faecal egg counts.
`postFEC`	numeric vector. Post-treatment faecal egg counts.
`rawCounts`	logical. If TRUE, `preFEC` and `postFEC` correspond to raw counts (as counted on equipment). Otherwise they correspond to calculated epgs (raw counts times correction factor). Defaults to `FALSE`.
`preCF`	a positive integer or a vector of positive integers. Pre-treatment correction factor(s).
`postCF`	a positive integer or a vector of positive integers. Post-treatment correction factor(s).
`paired`	logical. If TRUE, uses the model for the paired design. Otherwise uses the model for the unpaired design
`indEfficacy`	logical. If TRUE, uses the paired model allowing for individual efficacy. Only use in combination with `paired = TRUE` and `zeroInflation = FALSE`.
`zeroInflation`	logical. If TRUE, uses the model with zero-inflation. Otherwise uses the model without zero-inflation.
`muPrior`	a named list. Prior for the group mean epg parameter μ. The default prior is `list(priorDist = "gamma", hyperpars = c(1,0.001))`, i.e. a gamma distribution with shape 1 and rate 0.001, its 90% probability mass lies between 51 and 2996.
`kappaPrior`	a named list. Prior for the group dispersion parameter κ. The default prior is `list(priorDist = "gamma", hyperpars = c(1,0.7))`, i.e. a gamma distribution with shape 1 and rate 0.7, its 90% probability mass lies between 0.1 and 4.3 with a median of 1.
`deltaPrior`	a named list. Prior for the reduction parameter δ. The default prior is `list(priorDist = "beta", hyperpars = c(1,1))`, i.e. a uniform prior between 0 and 1.

phiPrior
: a named list. Prior for the zero-inflation parameter ϕ. The default prior is `list(priorDist = "beta", hyperpars = c(1,1))`, i.e. a uniform prior between 0 and 1.

deltakappaPrior
: a named list. Prior information for the shape parameter of reduction δ_κ. The default prior is `list(priorDist = "normal",hyperpars=c(2,1))`. Only used if `indEfficacy = TRUE`.

nsamples
: a positive integer. Number of samples for each chain (including burn-in samples).

nburnin
: a positive integer. Number of burn-in samples.

thinning
: a positive integer. Thinning parameter, i.e. the period for saving samples.

nchain
: a positive integer. Number of chains.

ncore
: a positive integer. Number of cores to use when executing the chains in parallel.

adaptDelta
: numeric. The target acceptance rate, a numeric value between 0 and 1.

saveAll
: logical. If TRUE, posterior samples for all parameters are saved in the `stanfit` object. If FALSE, only samples for δ, μ, κ and ϕ are saved. Default to FALSE.

verbose
: logical. If TRUE, prints progress and debugging information.

Details

List of built-in models:

- unpaired without zero-inflation: set `paired = FALSE`, `indEfficacy = FALSE`, `zeroInflation = FALSE`
- unpaired with zero-inflation: set `paired = FALSE`, `indEfficacy = FALSE`, `zeroInflation = TRUE`
- paired without zero-inflation: set `paired = TRUE`, `indEfficacy = FALSE`, `zeroInflation = FALSE`
- paired with zero-inflation: set `paired = TRUE`, `indEfficacy = FALSE`, `zeroInflation = TRUE`
- paired with individual efficacy: set `paired = TRUE`, `indEfficacy = TRUE`, `zeroInflation = FALSE`

Prior choice:

Consider using non-default prior for δ when,

- there is on average an increase in egg counts after treatment
- there are divergent-sample warnings
- there are non-convergence warnings

Two examples of useful non-default priors include:

1. `list(priorDist = "normal", hyperpars = c(1, 5))` for stablizing the reduction parameter without being informative.
2. `list(priorDist = "beta", hyperpars = c(0, 5))` for allowing up to 4-fold increase of egg count after treatment.

Other information: The first time each model with non-default priors is applied, it can take up to 20 seconds to compile the model. Currently the function only support prior distributions with two parameters. For a complete list of supported priors and their parameterization, please consult the list of distributions in Stan User Guide.

The default number of samples per chain is 2000, with 1000 burn-in samples. Normally this is sufficient in Stan. If the chains do not converge, one should tune the MCMC parameters until convergence is reached to ensure reliable results.

Value

Prints out the posterior summary of `FECR` as the reduction, `meanEPG.untreated` as the mean pre-treatment epg, and `meanEPG.treated` as the mean after-treatment epg. The posterior summary contains the mean, standard deviation (sd), 2.5%, 50% and 97.5% percentiles, the 95% highest posterior density interval (HPDLow95 and HPDHigh95) and the posterior mode.

NOTE: Based on our simulation studies, we recommend to use (2.5%, 97.5%) as the 95% credible interval and the median as summary statistics of reduction for the individual efficacy model. For all other models, we recommend to use the 95% HPD interval and the mode.

The returned value is a list that consists of:

`stan.samples` an object of S4 class `stanfit` representing the fitted results

`posterior.summary`
a data.frame that is the same as the printed posterior summary

Author(s)

Craig Wang

References

Individual efficacy models: Craig Wang, Paul R. Torgerson, Ray M. Kaplan, Melissa M. George, Reinhard Furrer. (2018) Modelling anthelmintic resistance by extending eggCounts package to allow individual efficacy, International Journal for Parasitology: Drugs and Drug Resistance, Volume 8, Pages 386-393. <doi:10.1016/j.ijpddr.2018.07.003>

Zero-inflation models: Craig Wang, Paul R. Torgerson, Johan Hoglund, Reinhard Furrer. (2017) Zero-inflated hierarchical models for faecal egg counts to assess anthelmintic efficacy, Veterinary Parasitology, Volume 235, Pages 20-28. <doi:10.1016/j.vetpar.2016.12.007>

Other models: Paul R. Torgerson, Michaela Paul, Reinhard Furrer. (2014) Evaluating faecal egg count reduction using a specifically designed package 'eggCounts' in R and a user friendly web interface, International Journal for Parasitology, Volume 44, Pages 299-303. <doi:10.1016/j.ijpara.2014.01.005>

See Also

`simData2s` for simulating faecal egg counts data with two samples

Examples

```
## load sample data
data(epgs)

## apply individual efficacy model to the data vectors
model <- fecr_stan(epgs$before, epgs$after, rawCounts = FALSE, preCF = 50,
                   paired = TRUE, indEfficacy = TRUE)
## convert to MCMC object and inspect the summary
samples <- stan2mcmc(model$stan.samples)
summary(samples)
```

`fecr_stanExtra` *Model the reduction of faecal egg count using custom models*

Description

Models the reduction in faecal egg counts with custom model formulation using Stan modelling language (for advanced users).

Usage

```
fecr_stanExtra(preFEC, postFEC, rawCounts = FALSE, preCF = 50, postCF = preCF,
  modelName = NULL, modelCode = NULL, modelFile = NULL, modelData = NULL,
  nsamples = 2000, nburnin = 1000, thinning = 1, nchain = 2,
  ncore = 1, adaptDelta = 0.95, verbose = FALSE)
```

Arguments

`preFEC`	numeric vector. Pre-treatment faecal egg counts. Not required if `modelData` is supplied.
`postFEC`	numeric vector. Post-treatment faecal egg counts. Not required if `modelData` is supplied.
`rawCounts`	logical. If TRUE, `preFEC` and `postFEC` correspond to raw counts (as counted on equipment). Otherwise they correspond to calculated epgs (raw counts times correction factor). Defaults to `FALSE`. Not required if `modelCode` or `modelFile` is supplied.
`preCF`	a positive integer or a vector of positive integers. Pre-treatment correction factor(s). Not required if `modelCode` or `modelFile` is supplied.
`postCF`	a positive integer or a vector of positive integers. Post-treatment correction factor(s). Not required if `modelCode` or `modelFile` is supplied.
`modelName`	string. One of four availale models ("Po", "UPo", "ZIPo", "ZIUPo") from **eggCountsExtra** package, which corresponds to outlier-adjusted version of paired, unpaired, paired with zero inflation and unpaired with zero inflation models. Not required if `modelCode` or `modelFile` is supplied.
`modelCode`	stan model code. Not required when `modelName` or `modelFile` is supplied.

`modelFile`	stan model file with file extension '*.stan'. Not required when `modelName` or `modelCode` is supplied.
`modelData`	stan data list. A named list or environment providing the data for the model, or a character vector for all the names of objects in the current enviroment used as data. Not required when `modelName` is supplied.
`nsamples`	a positive integer. Number of samples for each chain (including burn-in samples).
`nburnin`	a positive integer. Number of burn-in samples.
`thinning`	a positive integer. Thinning parameter, i.e. the period for saving samples.
`nchain`	a positive integer. Number of chains.
`ncore`	a positive integer. Number of cores to use when executing the chains in parallel.
`adaptDelta`	numeric. The target acceptance rate, a numeric value between 0 and 1.
`verbose`	logical. If TRUE, prints progress and debugging information.

Details

If `modelName` is one of c("Po", "UPo", "ZIPo", "ZIUPo"), then outlier-adjusted models are used.

- In paired models, outliers are those counts with postFEC > preFEC. Outlier weights are assigned as the inverse of postFEC/preFEC,
- In unpaired models, outliers are those counts with postFEC greater than the 95th percentile of a Poisson distribution, where the Poisson mean is computed based on the mean of postFEC excluding postFEC > Q3 + 1.5*IQR. Q3 is the 75th percentile and IQR is the interquartile range. The lowest outlier weight is assigned as 0.01, and other outliers assigned proportionally.
- In both cases, non-outliers are assigned with outlier weight = 1.

The first time each model is applied, it can take up to 20 seconds for Stan to compile the model.

The default number of samples per chain is 2000, with 1000 burn-in samples. Normally this is sufficient in Stan. If the chains do not converge, one should tune the MCMC parameters until convergence is reached to ensure reliable results.

Value

Prints out the posterior summary of `FECR` as the reduction, `meanEPG.untreated` as the mean pretreatment epg, and `meanEPG.treated` as the mean after-treatment epg. The posterior summary contains the mean, standard deviation (sd), 2.5%, 50% and 97.5% percentiles, the 95% highest posterior density interval (HPDLow95 and HPDHigh95) and the posterior mode.

The returned value is a list that consists of:

`stan.model`	an object of class `stanmodel-class` that was used
`stan.samples`	an object of S4 class `stanfit` representing the fitted results
`posterior.summary`	a data.frame that is the same as the printed posterior summary. Not available for custom models.

Author(s)

Craig Wang

Examples

```
## Not run:
library(eggCountsExtra)
data(epgs) ## load sample data

## apply paired model with outliers
model1 <- fecr_stanExtra(epgs$before, epgs$after, rawCounts=FALSE,
         preCF=10, modelName = "Po")
samples <- stan2mcmc(model1$stan.samples)
fecr_probs(model1$stan.samples, threshold = 0.99)

## apply a simple custom model
code <- "data{
  int J; // number of animals
  int y_before[J]; // after treatment McMaster count
  int y_after[J]; // before treatment McMaster count
}
parameters{
  real<lower=0> mu;
  real<lower=0,upper=1> delta;
}
model{
  mu ~ gamma(1,0.001);
  delta ~ beta(1,1);
  y_before ~ poisson(mu);
  y_after ~ poisson(mu*delta);
}"

dat <- list(J = nrow(epgs), y_before = epgs$before,
            y_after = epgs$after)
model2 <- fecr_stanExtra(modelCode = code, modelData = dat)

## End(Not run)
```

`fecr_stanSimple` *Model the reduction of faecal egg count using a simple Bayesian model*

Description

Models the reduction in faecal egg counts with a simple Bayesian model formulation. The model is for paired design only, and it assumes Poisson distribution for the observed egg counts.

Usage

```
fecr_stanSimple(preFEC, postFEC, rawCounts = FALSE,
  preCF = 50, postCF = preCF, muPrior, deltaPrior,
  nsamples = 2000, nburnin = 1000, thinning = 1, nchain = 2,
  ncore = 1, adaptDelta = 0.95, saveAll = FALSE, verbose = FALSE)
```

Arguments

`preFEC`	numeric vector. Pre-treatment faecal egg counts.
`postFEC`	numeric vector. Post-treatment faecal egg counts.
`rawCounts`	logical. If TRUE, `preFEC` and `postFEC` correspond to raw counts (as counted on equipment). Otherwise they correspond to calculated epgs (raw counts times correction factor). Defaults to `FALSE`.
`preCF`	positive integer or vector of positive integers. Pre-treatment correction factor(s).
`postCF`	positive integer or vector of positive integers. Post-treatment correction factor(s).
`muPrior`	named list. Prior for the group mean epg parameter μ. The default prior is `list(priorDist = "gamma",hyperpars=c(1,0.001))`, i.e. a gamma distribution with shape 1 and rate 0.001, its 90% probability mass lies between 51 and 2996.
`deltaPrior`	named list. Prior for the reduction parameter δ. The default prior is `list(priorDist = "beta",hyperpars=c(1,1))`, i.e. a uniform prior between 0 and 1.
`nsamples`	a positive integer. Number of samples for each chain (including burn-in samples).
`nburnin`	a positive integer. Number of burn-in samples.
`thinning`	a positive integer. Thinning parameter, i.e. the period for saving samples.
`nchain`	a positive integer. Number of chains.
`ncore`	a positive integer. Number of cores to use when executing the chains in parallel.
`adaptDelta`	numeric. The target acceptance rate, a numeric value between 0 and 1.
`saveAll`	logical. If TRUE, posterior samples for all parameters are saved in the `stanfit` object. Otherwise only samples for δ and μ are saved. Default to FALSE.
`verbose`	logical. If TRUE, prints progress and debugging information.

Details

The first time each model with non-default priors is applied, it can take up to 20 seconds to compile the model. Currently the function only support prior distributions with two parameters. For a complete list of supported priors and their parameterization, please consult the list of distributions in Stan User Guide.

The default number of samples per chain is 2000, with 1000 burn-in samples. Normally this is sufficient in Stan. If the chains do not converge, one should tune the MCMC parameters until convergence is reached to ensure reliable results.

Value

Prints out the posterior summary of FECR as the reduction, meanEPG.untreated as the mean pre-treatment epg, and meanEPG.treated as the mean after-treatment epg. The posterior summary contains the mean, standard deviation (sd), 2.5%, 50% and 97.5% percentiles, the 95% highest posterior density interval (HPDLow95 and HPDHigh95) and the posterior mode.

NOTE: we recommend to use the 95% HPD interval and the mode for further statistical analysis.

The returned value is a list that consists of:

`stan.samples` an object of S4 class `stanfit` representing the fitted results

`posterior.summary`
A data.frame that is the same as the printed posterior summary

Author(s)

Tea Isler
Craig Wang

See Also

`simData2s` for simulating faecal egg counts data with two samples

Examples

```
## load sample data
data(epgs)

## apply paired model with individual efficacy
model <- fecr_stanSimple(epgs$before, epgs$after,
            rawCounts = FALSE, preCF = 10)
samples <- stan2mcmc(model$stan.samples)
```

---

fec_stan *Modelling of faecal egg count data (one-sample case)*

Description

Models the mean of faecal egg counts with Bayesian hierarchical models. See Details for a list of model choices.

Usage

```
fec_stan(fec, rawCounts = FALSE, CF = 50, zeroInflation = TRUE,
  muPrior, kappaPrior, phiPrior, nsamples = 2000, nburnin = 1000,
  thinning = 1, nchain = 2, ncore = 1, adaptDelta = 0.95,
  saveAll = FALSE, verbose = FALSE)
```

Arguments

`fec`	numeric vector. Faecal egg counts.
`rawCounts`	logical. If TRUE, `preFEC` and `postFEC` correspond to raw counts (as counted on equipment). Otherwise they correspond to calculated epgs (raw counts times correction factor). Defaults to `FALSE`.
`CF`	a positive integer or a vector of positive integers. Correction factor(s).
`zeroInflation`	logical. If true, uses the model with zero-inflation. Otherwise uses the model without zero-inflation
`muPrior`	named list. Prior for the group mean epg parameter μ. The default prior is `list(priorDist = "gamma",hyperpars=c(1,0.001))`, i.e. a gamma distribution with shape 1 and rate 0.001, its 90% probability mass lies between 51 and 2996.
`kappaPrior`	named list. Prior for the group dispersion parameter κ. The default prior is `list(priorDist = "gamma",hyperpars=c(1,0.7))`, i.e. a gamma distribution with shape 1 and rate 0.7, its 90% probability mass lies between 0.1 and 4.3 with a median of 1.
`phiPrior`	named list. Prior for the zero-inflation parameter ϕ. The default prior is `list(priorDist = "beta",hyperpars=c(1,1))`, i.e. a uniform prior between 0 and 1.
`nsamples`	a positive integer. Number of samples for each chain (including burn-in samples).
`nburnin`	a positive integer. Number of burn-in samples.
`thinning`	a positive integer. Thinning parameter, i.e. the period for saving samples.
`nchain`	a positive integer. Number of chains.
`ncore`	a positive integer. Number of cores to use when executing the chains in parallel.
`adaptDelta`	numeric. The target acceptance rate, a numeric value between 0 and 1.
`saveAll`	logical. If TRUE, posterior samples for all parameters are saved in the `stanfit` object. If FALSE, only samples for μ, κ and ϕ are saved. Default to FALSE.
`verbose`	logical. If true, prints progress and debugging information.

Details

List of built-in models:

- without zero-inflation: set `zeroInflation = FALSE`
- with zero-inflation: set `zeroInflation = TRUE`

Note that this function only models the mean of egg counts, see `fecr_stan()` for modelling the reduction.

Other information: The first time each model with non-default priors is applied, it can take up to 20 seconds to compile the model. Currently the function only support prior distributions with two parameters. For a complete list of supported priors and their parameterization, please consult the list of distributions in Stan User Guide.

The default number of samples per chain is 2000, with 1000 burn-in samples. Normally this is sufficient in Stan. If the chains do not converge, one should tune the MCMC parameters until convergence is reached to ensure reliable results.

Value

Prints out summary of meanEPG as the posterior mean epg. The posterior summary contains the mean, standard deviation (sd), 2.5%, 50% and 97.5% percentiles, the 95% highest posterior density interval (HPDLow95 and HPDHigh95) and the posterior mode. NOTE: we recommend to use the 95% HPD interval and the mode for further statistical analysis.

The returned value is a list that consists of:

stan.samples	an object of S4 class stanfit representing the fitted results
posterior.summary	a data.frame that is the same as the printed posterior summary

Author(s)

Craig Wang

See Also

simData1s for simulating faecal egg count data with one sample

Examples

```
## load the sample data
data(epgs)

## apply zero-infation model
model <- fec_stan(epgs$before, rawCounts = FALSE, CF = 50)
```

---

getPrior_delta *Get prior parameters from Beta distribution*

Description

Compute the shape parameters from a Beta distribution for δ based on some prior belief.

Usage

```
getPrior_delta(lower, upper, p = 0.7, mode, conc, plot = TRUE)
```

Arguments

lower, upper, p	numeric. Prior belief about the reduction. There is p probability that the reduction is between lower and upper. Not used if mode and conc are supplied.
mode, conc	numeric. Prior belief about the reduction. The mode and concentration parameters of a beta distribution. Higher concentration indicates smaller variance. Not used if lower and upper thresholds are supplied.
plot	logical. If TRUE, the prior distribution is plotted after parameters are found.

Details

The multiroot function from **rootSolve** package is used to compute the parameters.

Value

Returns Beta prior parameters for δ and the printed argument to use in a fecr_stan() or a fec_stan() function call.

Author(s)

Tea Isler
Craig Wang

Examples

```
# there is 80% probability that the reduction is between 60% and 90%
getPrior_delta(lower = 0.6, upper = 0.9, p = 0.8)
```

---

getPrior_mu *Get prior parameters from Gamma distribution*

Description

Compute the shape and rate parameters from a Gamma distribution for μ based on some prior belief about its cumulative distribution function.

Usage

```
getPrior_mu(x, px, y, py, s1 = 1, s2 = 0.001, plot = TRUE)
```

Arguments

x, px, y, py	numeric. Threshold of some prior belief about true epg. There is px probability that the true epg is below x, and there is py probability that the true epg is below y.
s1, s2	numeric. Starting values.
plot	logical. If TRUE, the prior distribution is plotted after parameters are found.

Details

multiroot function from **rootSolve** package is used to compute the parameters.

Value

Returns Gamma prior parameters for μ and the printed argument to use in a fecr_stan() or a fec_stan() function call.

Author(s)

Tea Isler
Craig Wang

Examples

```
# there is 30% probability that the mean epg is less than 200
# and 80% probability that the mean epg is less than 500
getPrior_mu(x = 200, px = 0.3, y = 500, py = 0.8)
```

---

plotCounts *Plot faecal egg count data*

Description

Plot egg count data to reflect changes between before and after treatment.

Usage

```
plotCounts(data, paired = TRUE, points = TRUE,
    points.method = "jitter",  xlabel = "",
    ylabel = "Faecal egg counts [epg]", ...)
```

Arguments

`data`	a data.frame with two columns, the first column is before treatment counts, the second column is after treatment counts.
`paired`	logical. If TRUE, uses the plot for the paired design. Otherwise uses the plot for the unpaired design.
`points`	logical. If TRUE, add individual points for unpaired plot. Not used if `paired = TRUE`.
`points.method`	string. It is used to separate coincident points if `points = TRUE`. One of "jitter", "stack" or "overplot".
`xlabel`	string. Label of x-axis.
`ylabel`	String. Label of y-axis.
`...`	Additional arguments for function `xyplot` if paired is TRUE, for function `boxplot` otherwise.

Details

For paired data, a xyplot is used. For unpiared data, a grouped boxplot is used.

Value

A plot is drawn.

Author(s)

Craig Wang

Examples

```
data(epgs)
plotCounts(epgs[,c("before","after")], paired = TRUE)
```

---

simData1s — *Simulate faecal egg count data (1-sample situation)*

Description

Simulates (zero-inflated) egg count data

Usage

```
simData1s(n = 10, mean = 500, kappa = 0.5, phi = 1,
  f = 50, rounding = TRUE, seed = NULL)
```

Arguments

`n`	positive integer. Sample size.
`mean`	numeric. True number of eggs per gram (epg).
`kappa`	numeric. Overdispersion parameter, $\kappa \to \infty$ corresponds to Poisson distribution.
`phi`	numeric. Prevalence, i.e. proportion of infected animals, between 0 and 1.
`f`	positive integer. Correction factor of the egg counting technique, either an integer or a vector of integers with length `n`.
`rounding`	logical. If TRUE, the Poisson mean for the raw counts is rounded. The rounding applies since the mean epg is frequently reported as an integer value. For more information, see Details.
`seed`	integer. Random seed.

Details

In the simulation of raw (`master`) counts, it follows a Poisson distribution with some mean. The mean is frequently rounded down if it has a very low value and `rounding = TRUE`, hence there expects to be a bias overall when $\mu < 150$. Set `rounding = FALSE` for not to have any bias in the simulated counts.

Value

A data.frame with three columns, namely the observed epg (`obs`), actual number of eggs counted (`master`) and true epg in the sample (`true`).

Author(s)

Craig Wang
Michaela Paul

See Also

fec_stan for analyzing faecal egg count data with one sample

Examples

```
fec <- simData1s(n = 10, mean = 500,
          kappa = 0.5, phi = 0.7)
```

---

simData2s *Simulate faecal egg count data (2-sample situation)*

Description

Generates two samples of (zero-inflated) egg count data

Usage

```
simData2s(n = 10, preMean = 500, delta = 0.1, kappa = 0.5,
  deltaShape = NULL, phiPre = 1, phiPost = phiPre, f = 50,
  paired = TRUE, rounding = TRUE, seed = NULL)
```

Arguments

n	positive integer. Sample size.
preMean	numeric. True pre-treatment epg.
delta	numeric. Proportion of epg left after treatment, between 0 and 1. 1 - δ is reduction in mean after treatment, delta = 0.1 indicates a 90% reduction.
kappa	numeric. Overdispersion parameter, $\kappa \to \infty$ corresponds to Poisson distribution.
deltaShape	numeric. Shape parameter for the distribution of reductions. If NULL, the same reduction is applied to the latent true epg of each animal.
phiPre	numeric. Pre-treatment prevalence (i.e. proportion of infected animals), between 0 and 1.
phiPost	numeric. Post-treatment prevalence, between 0 and 1.
f	integer or vector of integers. Correction factor of the egg counting technique
paired	logical. If TRUE, paired samples are simulated. Otherwise unpaired samples are simulated.
rounding	logical. If TRUE, the Poisson mean for the raw counts is rounded. The rounding applies since the mean epg is frequently reported as an integer value. For more information, see Details.
seed	an integer that will be used in a call to set.seed before simulation. If NULL, a random seed is allocated.

Details

In the simulation of raw (`master`) counts, it follows a Poisson distribution with some mean. The mean is frequently rounded down if it has a very low value and `rounding = TRUE`, there expects to be some bias in the mean reduction when $\mu < 150$ and $\delta < 0.1$. Set `rounding = FALSE` for not to have any bias.

Value

A data.frame with six columns, namely the observed epg (`obs`), actual number of eggs counted (`master`) and true epg in the sample (`true`) for both pre- and post- treatment.

Author(s)

Craig Wang
Michaela Paul

See Also

`fecr_stan` for analyzing faecal egg count data with two samples

Examples

```
fec <- simData2s(n = 10, preMean = 500, delta = 0.1, kappa = 0.5)

## show the bias when the true reduction should be 95%
fec <- simData2s(n = 1e5, preMean = 150, delta = 0.05,
          kappa = 0.5, seed = 1)
1 - mean(fec$masterPost)/mean(fec$masterPre)
## without bias
fec <- simData2s(n = 1e5, preMean = 150, delta = 0.05,
          kappa = 0.5, seed = 1, rounding = FALSE)
1 - mean(fec$masterPost)/mean(fec$masterPre)
```

---

stan2mcmc — *Convert a Stanfit object to a MCMC object*

Description

Converts a `stanfit` object into a `mcmc` object for easier analysis.

Usage

```
stan2mcmc(stanFit)
```

Arguments

`stanFit`	a `stanfit` object from the output of either `fecr_stan()` or `fec_stan()`

Details

The output can be analyzed as a mcmc object with the functions from the coda package. NOTE: The resulting MCMC object does not contain warm-up samples and is already thinned.

Value

A MCMC object with a list of relevant parameters.

Author(s)

Craig Wang

Examples

```
data(epgs)

## apply zero-infation model for the paired design
model <- fecr_stan(epgs$before, epgs$after, rawCounts = FALSE,
            indEfficacy = FALSE, preCF = 10,
            paired = TRUE, zeroInflation = TRUE)
samples <- stan2mcmc(model$stan.samples)
summary(samples)
plot(samples)
```

Index