

Package ‘brr’

October 12, 2022

Type Package

Title Bayesian Inference on the Ratio of Two Poisson Rates

Version 1.0.0

Date 2015-09-07

Author Stéphane Laurent

Maintainer Stéphane Laurent <laurent_step@yahoo.fr>

Description Implementation of the Bayesian inference for the two independent Poisson samples model, using the semi-conjugate family of prior distributions.

Depends R (>= 3.0.0)

License GPL-2

Imports gsl, SuppDists, TeachingDemos, hypergeo, pander, stringr, methods

Suggests testthat, R.rsp, magrittr

VignetteBuilder R.rsp

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2015-09-07 18:17:47

R topics documented:

brr-package	2
Beta2Dist	3
BNBDist	4
Brr	5
FrequentistInference	6
GB2Dist	7
GIBDist	9
Inference	10
inference.brr	11
Intrinsic2Inference	12

intrinsic2_discrepancy	13
IntrinsicInference	14
intrinsic_discrepancy	15
PGB2Dist	16
PGIBDist	17
plot.brr	18
Posterior_lambda	19
Posterior_mu	20
Posterior_phi	21
Post_x	22
Post_y	23
PriorAndPosterior	24
Prior_lambda	26
Prior_mu	27
Prior_phi	28
Prior_x	29
Prior_x_given_y	30
Prior_y	31
summary_gamma	32
summary_nbinom	33

Index	34
--------------	-----------

brr-package

Bayesian inference on the ratio of two Poisson rates

Description

Implementation of the Bayesian inference for the rate ratio of two independent Poisson samples, using the semi-conjugate family of prior distributions, and a default non-informative prior.

To learn more, start with the vignettes: `browseVignettes(package="brr")`.

If you encounter a bug, or if you have a suggestion to improve the package, please file an issue on the github repo <https://github.com/stla/brr>.

Details

Package: brr
 Type: Package
 Version: 1.0.0
 Date: 2015-09-07
 License: GPL-2

Author(s)

Stéphane Laurent

References

S. Laurent, C. Legrand (2012): A Bayesian framework for the ratio of two Poisson rates in the context of vaccine efficacy trials. *ESAIM, Probability & Statistics* 16 (2012), 375–398.

S. Laurent (2012): Some Poisson mixtures distributions with a hyperscale parameter. *Brazilian Journal of Probability and Statistics* 26 (2012), 265–278.

S. Laurent (2012): Intrinsic Bayesian inference on a Poisson rate and on the ratio of two Poisson rates. *Journal of Statistical Planning and Inference* 142 (2012), 2656–2671.

Beta2Dist

Beta distribution of the second kind

Description

Density, distribution function, quantile function and random generation for the Beta distribution of the second kind with shape parameters c and d and scale parameter $scale$.

Usage

```
dbeta2(x, c, d, scale, log = FALSE, ...)  
pbeta2(q, c, d, scale, ...)  
qbeta2(p, c, d, scale, ...)  
rbeta2(n, c, d, scale)  
summary_beta2(c, d, scale, output = "list", ...)
```

Arguments

<code>x, q</code>	vector of quantiles
<code>c, d</code>	non-negative shape parameters
<code>scale</code>	non-negative scale parameter
<code>log</code>	logical; if true, returns the logarithm of the result
<code>...</code>	other arguments passed to FDist
<code>p</code>	vector of probabilities
<code>n</code>	number of observations to be simulated
<code>output</code>	type of the <code>summary_beta2</code> output: "list" to return a list, "pandoc" to print a table

Details

The Beta distribution of the second kind with shape parameters $c > 0$ and $d > 0$ and scale parameter $k > 0$ is the distribution of $k * (U/(1 - U))$ where U is a random variable following the Beta distribution with shape parameters c and d .

Value

dbeta2 gives the density, pbeta2 the distribution function, qbeta2 the quantile function, and rbeta2 generates random observations, and summary_beta2 returns a summary of the distribution.

Note

Beta2Dist is a generic name for the functions documented.

Examples

```
curve(dbeta2(x, 3, 10, scale=2), from=0, to=3)
u <- rbeta(1e5, 3, 10)
lines(density(2*u/(1-u)), col="blue", lty="dashed")
summary_beta2(3,10,2)
```

 BNBDist

Beta-negative binomial distribution

Description

Density, cumulative function, quantile function and random generation for the Beta-negative binomial distribution with shape parameters a , c , d .

Usage

```
dbeta_nbinom(x, a, c, d, ...)
pbeta_nbinom(q, a, c, d, ...)
qbeta_nbinom(p, a, c, d, ...)
rbeta_nbinom(n, a, c, d)
sbeta_nbinom(a, c, d)
summary_beta_nbinom(a, c, d, output = "list", ...)
```

Arguments

<code>x, q</code>	vector of non-negative integer quantities
<code>a, c, d</code>	non-negative shape parameters
<code>...</code>	other arguments passed to ghyper
<code>p</code>	vector of probabilities
<code>n</code>	number of observations to be sampled
<code>output</code>	type of the <code>summary_beta_nbinom</code> output: "list" to return a list, "pandoc" to print a table

Details

This is the mixture distribution obtained by sampling a value b from a Beta distribution with parameters c, d , then sampling a value λ from a Gamma distribution with shape a and rate $b/(1 - b)$, and then sampling a Poisson distribution with mean λ .

Value

`dbeta_nbinom` gives the density, `pbeta_nbinom` the cumulative function, `qbeta_nbinom` the quantile function, `rbeta_nbinom` samples from the distribution, `sbeta_nbinom` and `summary_beta_nbinom` give some summaries of the distribution.

Note

BNBDist is a generic name for the functions documented.

Examples

```
a <- 2 ; c <- 5 ; d <- 30
barplot(dbeta_nbinom(0:50, a, c, d), names=0:50)
summary_beta_nbinom(a, c, d)
```

Brr

Creation and summary of a brr object

Description

Set up the Bayesian model and the observations

Usage

```
Brr(...)
```

```
## S3 method for class 'brr'
summary(object, phi0 = 1, hypothesis = "greater", ...)
```

```
## S3 method for class 'summary.brr'
print(x, table.style = "grid", ...)
```

Arguments

...	prior parameters a, b, c, d, samples sizes S, T, observed counts x, y, future sample sizes Snew, Tnew, to be set as in a list (see examples)
object	an object of class brr
phi0	the value of interest of the rate ratio
hypothesis	"greater" to return $Pr(\phi > \phi_0)$, "lower" to return $Pr(\phi < \phi_0)$
x	the output to be printed
table.style	the style of the table to print (passed to <code>pandoc.table.return</code>)

Value

Brr returns an object of class brr, `summary.brr` returns a list but prints its contents through `print.summary.brr`

Examples

```

model <- Brr(a=2, b=3)
model()
# add parameters
model <- model(c=4, d=5)
model()
# replace parameters
model <- model(a=10, b=11)
model()
model <- Brr()
summary(model)
model <- Brr(x=3, y=4)
summary(model)
model <- Brr(a=2, b=4, T=10)
summary(model)
model <- model(a=2, b=4, c=3, d=5, S=10, T=10)
summary(model)
model <- model(x=5, y=10)
summary(model)

```

Description

Frequentist confidence intervals about the relative risk: binomial interval (`rr_interval_binomial`) and Sahai and Khurshid confidence interval (`rr_interval_SK`)

Usage

```
rr_interval_SK(x, y, S, T, conf = 0.95)
rr_interval_binomial(x, y, S, T, conf = 0.95)
rr_intervals(x, y, S, T, conf = 0.95)
```

Arguments

x,y	Observed counts
S,T	sample sizes
conf	confidence level

Details

The binomial interval (`rr_interval_binomial`) is the classical confidence interval obtained by conditioning on the sum $x+y$ of the two counts. The same interval is implemented in the `rateratio.test` package. The Sahai and Khurshid interval (`rr_interval_SK`) is an unconditional confidence interval. See the reference for more details and a study of its performance.

Value

`rr_interval_binomial` and `rr_interval_SK` return the bounds of the confidence interval in a vector, `rr_intervals` returns a list with the two confidence intervals

References

S. Laurent, C. Legrand: A Bayesian framework for the ratio of two Poisson rates in the context of vaccine efficacy trials. *ESAIM, Probability & Statistics* 16 (2012), 375–398.

Examples

```
x <- 3; y <- 10; S <- 100; T <- 100
rr_intervals(x, y, S, T)
brr_intervals(x, y, S, T)
```

GB2Dist

Gamma-Beta2 distribution

Description

Density and random generation for the Gamma-Beta2 distribution with shape parameters a , c , d and rate parameter τ (scale of the Beta2 distribution).

Usage

```

dGB2(x, a, c, d, tau)

pGB2(q, a, c, d, tau, ...)

qGB2(p, a, c, d, tau)

rGB2(n, a, c, d, tau)

moment_GB2(k, a, c, d, tau)

summary_GB2(a, c, d, tau, output = "list", ...)

```

Arguments

<code>x, q</code>	vector of non-negative quantiles
<code>a, c, d</code>	non-negative shape parameters
<code>tau</code>	non-negative rate parameter
<code>...</code>	arguments passed to genhypergeo function
<code>p</code>	vector of probabilities
<code>n</code>	number of observations to be sampled
<code>k</code>	the order of the moment
<code>output</code>	type of the <code>summary_GB2</code> output: "list" to return a list, "pandoc" to print a table

Details

This is the mixture distribution obtained by sampling a value y from the [Beta2 distribution](#) with shape parameters c , d , and scale τ and then sampling a value from the Gamma distribution with shape a and rate y . The pdf involves the Kummer confluent hypergeometric function of the second kind. The cdf involves the generalized hypergeometric function. Its current implementation does not work when $a-d$ is an integer, and also fails for many other cases.

Value

`dGB2` gives the density, `pGB2` the cumulative function, `rGB2` samples from the distribution, and `summary_GB2` gives a summary of the distribution.

Note

GB2Dist is a generic name for the functions documented.

Examples

```

a <- 2 ; c <- 4 ; d <- 3 ; tau <- 1.67
sims <- rGB2(1e6, a, c, d, tau)
mean(sims); moment_GB2(1,a,c,d,tau)

```



```
mean(sims^2); moment_GB2(2,a,c,d,tau)
summary_GB2(a,c,d,tau)
```

GIBDist

Gamma-Inverse Beta distribution

Description

Density and random generation for the Gamma-Inverse Beta distribution with shape parameters a , α , β and rate parameter ρ .

Usage

```
dGIB(x, a, alpha, beta, rho)

rGIB(n, a, alpha, beta, rho)

summary_GIB(a, alpha, beta, rho, output = "list", ...)
```

Arguments

<code>x</code>	vector of quantiles
<code>a</code>	non-negative shape parameter of the Gamma distribution
<code>alpha, beta</code>	non-negative shape parameters of the mixing Beta distribution
<code>rho</code>	rate parameter
<code>n</code>	number of observations to be simulated
<code>output</code>	type of the <code>summary_GIB</code> output: "list" to return a list, "pandoc" to print a table
<code>...</code>	arguments passed to <code>pander.data.frame</code>

Details

This is the mixture distribution obtained by sampling a value b from a Beta distribution with shape parameters β , α and then sampling a Gamma distribution with shape a and rate ρ/b .

Value

`dGIB` gives the density, `rGIB` samples from the distribution, and `summary_GIB` returns a summary of the distribution.

Note

GIBDist is a generic name for the functions documented.

Examples

```
curve(dGIB(x, 3, 4, 2, 2.5), from=0, to=3)
summary_GIB(3, 4, 2, 2.5, output="pandoc", style="grid")
```

 Inference

Inference summaries

Description

Credibility intervals, estimates

Usage

```
brr_intervals(x, y, S, T, a = 0.5, b = 0, c = 0.5, d = 0,
             level = 0.95, intervals = "equi-tailed*", ...)
```

```
brr_estimates(x, y, S, T, a = 0.5, b = 0, c = 0.5, d = 0,
             parameter = "phi", ...)
```

Arguments

x,y	Observed counts
S,T	sample sizes
a,b,c,d	Prior parameters
level	confidence level
intervals	a character vector, the intervals to be returned
...	arguments passed to IntrinsicInference and Intrinsic2Inference
parameter	parameter of interest "phi" or "VE" (=1-phi)

Value

A list of confidence intervals (brr_intervals) or estimates (brr_estimates)

Note

Inference is a generic name for the functions documented.

See Also

[confint.brr](#)

Examples

```
brr_intervals(x=4, y=5, S=10, T=10, a=0.5, b=0, c=0.5, d=0)
brr_intervals(x=4, y=5, S=10, T=10, a=0.5, b=0, c=0.5, d=0, intervals=c("left","equi-tailed"))
brr_estimates(x=4, y=5, S=10, T=10, a=0.5, b=0, c=0.5, d=0)
```

inference.brr *Credibility intervals and estimates*

Description

Get credibility intervals and estimates from a brr object

Usage

```
## S3 method for class 'brr'  
confint(object, parm = NULL, level = 0.95,  
         intervals = "all", ...)  
  
## S3 method for class 'confint.brr'  
print(x, style = "grid", ...)  
  
## S3 method for class 'brr'  
coef(object, parameter = "phi", ...)  
  
## S3 method for class 'coef.brr'  
print(x, ...)  
  
## S3 method for class 'brr'  
predict(object, level = 0.95, ...)  
  
## S3 method for class 'predict.brr'  
print(x, style = "grid", ...)
```

Arguments

object	a brr object
parm	ignored
level	confidence level
intervals	a character vector, the intervals to be returned
...	other arguments passed to brr_intervals or brr_estimates
x	the output to be printed
style	the style of the table to print (passed to pandoc.table.return)
parameter	parameter of interest "phi" or "VE" (=1-phi)

Details

`confint.brr` is a wrapper to [brr_intervals](#) and `coef.brr` is a wrapper to [brr_estimates](#)

Value

`confint.brr` returns a list of confidence intervals, `coef.brr` returns a list of estimates, `predict.brr` returns a data frame.

Examples

```
model <- Brr(x=10, y=10, S=100, T=100)
confint(model)
coef(model)
predict(model)
predict(model, Snew=1000, Tnew=1000)
model <- model(Snew=1000, Tnew=1000)
predict(model)
```

Intrinsic2Inference *Intrinsic inference on the rates ratio based on the second intrinsic discrepancy.*

Description

Intrinsic inference on the rates ratio based on the second intrinsic discrepancy.

Usage

```
intrinsic2_phi0(phi0, x, y, S, T, a, b, c = 0.5, d = 0, beta_range = TRUE,
  tol = 1e-08, ...)

intrinsic2_phi0_sims(phi0, x, y, S, T, a, b, c = 0.5, d = 0,
  nsims = 1e+06)

intrinsic2_estimate(x, y, S, T, a, b, c = 0.5, d = 0, otol = 1e-08, ...)

intrinsic2_H0(phi.star, alternative, x, y, S, T, a, b, c = 0.5, d = 0, ...)

intrinsic2_bounds(x, y, S, T, a, b, c = 0.5, d = 0, conf = 0.95,
  parameter = "phi", otol = 1e-08, ...)
```

Arguments

<code>phi0</code>	the proxy value of ϕ
<code>x,y</code>	Observed counts
<code>S,T</code>	sample sizes
<code>a,b,c,d</code>	Prior parameters
<code>beta_range</code>	logical, if TRUE (default), an internal method is used to avoid a possible failure in numerical integration; see the main vignette for details
<code>tol</code>	accuracy requested

...	arguments passed to <code>integrate</code>
<code>nsims</code>	number of simulations
<code>otol</code>	desired accuracy for optimization
<code>phi.star</code>	the hypothesized value of ϕ
<code>alternative</code>	alternative hypothesis, "less" for $H_1: \phi_0 < \phi.star$, "greater" for $H_1: \phi_0 > \phi.star$
<code>conf</code>	credibility level
<code>parameter</code>	parameter of interest: relative risk "phi" or vaccine efficacy "VE"

Value

`intrinsic2_phi0` returns the posterior expected loss, `intrinsic2_estimate` returns the intrinsic estimate, `intrinsic2_H0` performs intrinsic hypothesis testing, and `intrinsic2_bounds` returns the intrinsic credibility interval.

Examples

```
a<-2; b<-10; c<-1/2; d<-0; S<-100; T<-S; x<-0; y<-20
intrinsic2_phi0(0.5, x, y, S, T, a, b, c, d)
intrinsic2_phi0_sims(0.5, x, y, S, T, a, b, c, d)
intrinsic2_estimate(x, y, S, T, a, b, c, d)
bounds <- intrinsic2_bounds(x, y, S, T, a, b, c, d, conf=0.95); bounds
ppost_phi(bounds[2], a, b, c, d, S, T, x, y) - ppost_phi(bounds[1], a, b, c, d, S, T, x, y)
```

`intrinsic2_discrepancy`

Second intrinsic discrepancy

Description

Intrinsic discrepancy from ϕ_0 to ϕ in the marginal model.

Usage

```
intrinsic2_discrepancy(phi0, phi, a, b, S, T)
```

Arguments

<code>phi0</code>	the proxy value of ϕ
<code>phi</code>	the true value of the parameter
<code>a, b,</code>	the parameters of the prior Gamma distribution on μ
<code>S, T</code>	sample sizes

Value

A number, the intrinsic discrepancy from ϕ_0 to ϕ .

IntrinsicInference *Intrinsic inference on the rate ratio.*

Description

Intrinsic inference on the rate ratio.

Usage

```
intrinsic_phi0(phi0, x, y, S, T, a = 0.5, b = 0, c = 0.5, d = 0,
  beta_range = TRUE, tol = 1e-08, ...)
```

```
intrinsic_phi0_sims(phi0, x, y, S, T, a = 0.5, b = 0, c = 0.5, d = 0,
  nsims = 1e+06)
```

```
intrinsic_estimate(x, y, S, T, a = 0.5, b = 0, c = 0.5, d = 0,
  otol = 1e-08, ...)
```

```
intrinsic_H0(phi.star, alternative, x, y, S, T, a = 0.5, b = 0, c = 0.5,
  d = 0, ...)
```

```
intrinsic_bounds(x, y, S, T, a = 0.5, b = 0, c = 0.5, d = 0,
  conf = 0.95, parameter = "phi", otol = 1e-08, ...)
```

Arguments

phi0	the proxy value of phi
x,y	Observed counts
S,T	sample sizes
a,b,c,d	Prior parameters
beta_range	logical, if TRUE (default), an internal method is used to avoid a possible failure in numerical integration; see the main vignette for details
tol	accuracy requested
...	other arguments passed to integrate
nsims	number of simulations
otol	desired accuracy for optimization
phi.star	the hypothesized value of phi
alternative	alternative hypothesis, "less" for $H_1: \phi_0 < \phi.star$, "greater" for $H_1: \phi_0 > \phi.star$
conf	credibility level
parameter	parameter of interest: relative risk "phi" or vaccine efficacy "VE"

Value

`intrinsic_phi0` returns the posterior expected loss, `intrinsic_estimate` returns the intrinsic estimate, `intrinsic_H0` performs intrinsic hypothesis testing, and `intrinsic_bounds` returns the intrinsic credibility interval.

Examples

```
a<-0.5; b<-0; c<-1/2; d<-0; S<-100; T<-S; x<-0; y<-20
intrinsic_phi0(0.5, x, y, S, T, a, b, c, d)
intrinsic_phi0_sims(0.5, x, y, S, T, a, b, c, d)
intrinsic_estimate(x, y, S, T, a, b, c, d)
bounds <- intrinsic_bounds(x, y, S, T, a, b, c, d, conf=0.95); bounds
ppost_phi(bounds[2], a, b, c, d, S, T, x, y) - ppost_phi(bounds[1], a, b, c, d, S, T, x, y)
```

`intrinsic_discrepancy` *Intrinsic discrepancy*

Description

Intrinsic discrepancy from ϕ_0 to (μ, ϕ) .

Usage

```
intrinsic_discrepancy(phi0, mu, phi, S, T)
```

Arguments

<code>phi0</code>	the proxy value of ϕ
<code>mu, phi</code>	the true values of the parameters
<code>S, T</code>	sample sizes

Value

A number, the intrinsic discrepancy from ϕ_0 to (μ, ϕ) .

PGB2Dist

Poisson-Gamma-Beta2 distribution

Description

Density and random generation for the Poisson-Gamma-Beta2 distribution with shape parameters a , c , d and hyperrate parameter τ (scale of the Beta2 distribution). For $\tau=1$ this is the same as the [Beta-negative binomial distribution](#).

Usage

```
dPGB2(x, a, c, d, tau)
pPGB2(q, a, c, d, tau)
qPGB2(p, a, c, d, tau)
rPGB2(n, a, c, d, tau)
summary_PGB2(a, c, d, tau, output = "list")
```

Arguments

x, q	vector of non-negative integer quantiles
a, c, d	non-negative shape parameters
τ	non-negative hyperrate parameter
p	vector of probabilities
n	number of observations to be sampled
output	type of the summary_PGB2 output: "list" to return a list, "pandoc" to print a table

Details

This is the mixture distribution obtained by sampling a value y from the [Beta2 distribution](#) with shape parameters c , d , and scale τ , then sampling a value λ from the Gamma distribution with shape a and rate y , and then sampling the Poisson distribution with mean λ .

Value

dPGB2 gives the density, pPGB2 the cumulative function, rPGB2 samples from the distribution, and summary_PGB2 gives a summary of the distribution.

Note

PGB2Dist is a generic name for the functions documented.

Examples

```
a <- 2 ; c <- 5 ; d <- 30; tau <- 2
barplot(dPGB2(0:40, a, c, d, tau), names=0:40)
summary_PGB2(a,c,d,tau, output="pandoc")
```

PGIBDist

*Poisson-Gamma-Inverse Beta distribution***Description**

Density and random generation for the Poisson-Gamma-Inverse Beta distribution with shape parameters a , c , d and scale parameter ρ .

Usage

```
dPGIB(x, a, alpha, beta, rho)
pPGIB(q, a, alpha, beta, rho)
qPGIB(p, a, alpha, beta, rho)
rPGIB(n, a, alpha, beta, rho)
summary_PGIB(a, alpha, beta, rho, output = "list", ...)
```

Arguments

x, q	vector of integer quantiles
a	non-negative shape parameter of the Gamma distribution
α, β	non-negative shape parameters of the mixing Beta distribution
ρ	hyperrate parameter (rate of the mixing distribution)
p	vector of probabilities
n	number of observations to be simulated
output	type of the summary_PGIB output: "list" to return a list, "pandoc" to print a table
...	arguments passed to pander.data.frame

Details

This is the mixture distribution obtained by sampling a value from a [Gamma-Inverse Beta distribution](#) and then sampling from a Poisson distribution having this value as mean.

Value

dPGIB gives the density, rPGIB samples from the distribution, and summary_PGIB gives a summary of the distribution.

Note

PGIBDist is a generic name for the functions documented.

Examples

```
barplot(dPGIB(0:5, a=13, alpha=4, beta=2, rho=2.5), names=0:5)
summary_PGIB(13, 4, 2, 2.5)
```

plot.brr

plot brr

Description

plot brr

Usage

```
## S3 method for class 'brr'
plot(x, what = "summary", bounds = NULL, ...)
```

Arguments

x	an object of class brr (see Brr)
what	"summary" to plot automatically the priors on mu and phi and the posterior on phi, or an expression like <code>dprior(mu)</code> for a specific plot (see examples)
bounds	for specific plot only, the range over which the function will be plotted; NULL for automatic bounds
...	other arguments passed to plot or barplot

Examples

```
model <- Brr(a=2, b=3)
plot(model)
plot(model, dprior(mu))
plot(model, dprior(mu), xlim=c(0,4), lwd=3, col="blue")
plot(model, pprior(mu))
plot(model, qprior(mu))
model <- model(c=4, d=6, S=10, T=10)
plot(model)
plot(model, dprior(phi))
plot(model, dprior(x))
model <- model(y=4)
plot(model, dprior(x_given_y))
model <- model(x=5, y=5)
plot(model, dpost(phi))
model <- model(Snew=10, Tnew=10)
plot(model, dpost(x))
```

Posterior_lambda	<i>Posterior distribution on the incidence rate in the treated group</i>
------------------	--

Description

Density and random generation for the posterior distribution on the rate in the treated group. The distribution function and the quantile function are not available.

Usage

```
dpost_lambda(lambda, a, c, d, S, x, y, ...)
```

```
rpost_lambda(n, a, c, d, S, x, y)
```

```
spost_lambda(a, c, d, S, x, y, ...)
```

Arguments

lambda	vector of quantiles
a	non-negative shape parameter of the Gamma prior distribution on μ
c, d	non-negative shape parameters of the prior distribution on ϕ
S	sample size in treated group
x, y	counts in the treated group and control group
...	other arguments passed to GIBDist
n	number of observations to be simulated

Details

The pdf of the posterior distribution of the incidence rate λ involves the Kummer confluent hypergeometric function of the second kind.

Value

dpost_lambda gives the density, and rpost_lambda samples from the distribution, and spost_lambda gives a summary of the distribution.

Note

Posterior_lambda is a generic name for the functions documented.

Examples

```
curve(dpost_lambda(x, 2, 2, 2, 20, 1, 10), from=0, to=0.4)
spost_lambda(2, 2, 2, 20, 1, 10)
```

 Posterior_mu

Posterior distribution on the rate in the control group

Description

Density and random generation for the posterior distribution on the rate in the control group. The distribution function and the quantile function are not available.

Usage

```
dpost_mu(mu, a, b, c, d, T, x, y, ...)
```

```
rpost_mu(n, a, b, c, d, T, x, y)
```

```
spost_mu(a, b, c, d, T, x, y, ...)
```

Arguments

mu	vector of quantiles
a,b	non-negative shape and rate parameter of the Gamma prior distribution on μ
c,d	non-negative shape parameters of the prior distribution on ϕ
T	sample size in control group
x,y	counts in the treated group and control group
...	other arguments passed to GIBDist
n	number of observations to be simulated

Details

The pdf of the posterior distribution of the incidence rate μ involves the Kummer confluent hypergeometric function of the second kind.

Value

dpost_mu gives the density, rpost_mu samples from the distribution, and spost_mu gives a summary of the distribution.

Note

Posterior_mu is a generic name for the functions documented.

Examples

```
curve(dpost_mu(x, 2, 2, 2, 2, 10, 3, 8), from=0, to=2)
spost_mu(2, 2, 2, 2, 10, 3, 8, output="pandoc")
```

 Posterior_phi

Posterior distribution on the relative risk and the vaccine efficacy

Description

Density, distribution function, quantile function and random generation for the posterior distribution on relative risk or the vaccine efficacy.

Usage

dpost_phi(phi, a, b, c, d, S, T, x, y, ...)

dpost_VE(VE, a, b, c, d, S, T, x, y, ...)

ppost_phi(q, a, b, c, d, S, T, x, y, ...)

ppost_VE(q, a, b, c, d, S, T, x, y, ...)

qpost_phi(p, a, b, c, d, S, T, x, y, ...)

qpost_VE(p, a, b, c, d, S, T, x, y, ...)

rpost_phi(n, a, b, c, d, S, T, x, y)

spost_phi(a, b, c, d, S, T, x, y, ...)

Arguments

phi, VE, q	vector of quantiles
a, b	non-negative shape parameter and rate parameter of the prior Gamma distribution on the control incidence rate
c, d	non-negative shape parameters of the prior distribution on the relative risk
S, T	sample sizes in control group and treated group
x, y	counts in the treated group and control group
...	other arguments passed to Beta2Dist
p	vector of probabilities
n	number of observations to be simulated

Details

The prior distribution on the relative risk ϕ is the Beta2 distribution with shape parameters c and d and scale parameter $(T + b)/S$.

Value

dpost_phi gives the density, ppost_phi the distribution function, qpost_phi the quantile function, rpost_phi samples from the distribution, and spost_phi gives a summary of the distribution.

Note

Posterior_phi is a generic name for the functions documented.

Examples

```
a <- 2; b <- 2; c <- 3; d <- 4; S <- 1; T <- 1; x <- 2; y <- 6
spost_phi(a, b, c, d, S, T, x, y, output="pandoc")
require(magrittr)
phi <- seq(0, 6, length.out=100)
phi %>% { plot(., dpost_phi(., a, b, c, d, S, T, x, y), type="l") }
phi %>% { lines(., dprior_phi(., b, c, d, S, T), col="red") }
```

 Post_x

Posterior predictive distribution of the count in the treated group

Description

Density, distribution function, quantile function and random generation for the posterior predictive distribution of the count in the treated group.

Usage

```
dpost_x(xnew, Snew, a = 0.5, c = 0.5, d = 0, x, y, S)
```

```
ppost_x(q, Snew, a = 0.5, c = 0.5, d = 0, x, y, S)
```

```
qpost_x(p, Snew, a = 0.5, c = 0.5, d = 0, x, y, S)
```

```
rpost_x(n, Snew, a = 0.5, c = 0.5, d = 0, x, y, S)
```

```
spost_x(Snew, a = 0.5, c = 0.5, d = 0, x, y, S, ...)
```

Arguments

xnew, q	vector of non-negative integer quantiles
a	non-negative shape parameter of the Gamma prior distribution on the rate μ
c, d	non-negative shape parameters of the prior distribution on ϕ
x, y	counts (integer) in the treated group and control group of the observed experiment
S, Snew	sample sizes of the treated group in the observed experiment and the predicted experiment

p vector of probabilities
 n number of observations to be simulated
 ... arguments passed to [summary_PGIB](#)

Details

The posterior predictive distribution of the count in the treated group is a [Poisson-Gamma-Inverse Beta distribution](#).

Value

dpost_x gives the density, ppost_x the distribution function, qpost_x the quantile function, rpost_x samples from the distribution, and spost_x gives a summary of the distribution.

Note

Post_x is a generic name for the functions documented.

Examples

```
barplot(dpost_x(0:10, 10, 2, 3, 4, 5, 3, 10))
qpost_x(0.5, 10, 2, 3, 4, 5, 3, 10)
ppost_x(4, 10, 2, 3, 4, 5, 3, 10)
```

Post_y

Posterior predictive distribution of the count in the control group

Description

Density, distribution function, quantile function and random generation for the posterior predictive distribution of the count in the control group.

Usage

```
dpost_y(ynew, Tnew, a = 0.5, b = 0, c = 0.5, d = 0, x, y, T)
ppost_y(q, Tnew, a = 0.5, b = 0, c = 0.5, d = 0, x, y, T)
qpost_y(p, Tnew, a = 0.5, b = 0, c = 0.5, d = 0, x, y, T)
rpost_y(n, Tnew, a = 0.5, b = 0, c = 0.5, d = 0, x, y, T)
spost_y(Tnew, a = 0.5, b = 0, c = 0.5, d = 0, x, y, T, ...)
```

Arguments

ynew, q	vector of non-negative integer quantiles
a, b	non-negative shape parameter and rate parameter of the Gamma prior distribution on the rate μ
c, d	non-negative shape parameters of the prior distribution on ϕ
x, y	counts (integer) in the treated group and control group of the observed experiment
T, Tnew	sample sizes of the control group in the observed experiment and the predicted experiment
p	vector of probabilities
n	number of observations to be simulated
...	arguments passed to summary_PGIB

Details

The posterior predictive distribution of the count in the treated group is a [Poisson-Gamma-Inverse Beta distribution](#).

Value

dpost_y gives the density, ppost_y the distribution function, qpost_y the quantile function, rpost_y samples from the distribution, and spost_y gives a summary of the distribution.

Note

Post_y is a generic name for the functions documented.

Examples

```
barplot(dpost_y(0:10, 10, 2, 7, 3, 4, 5, 3, 10))
spost_y(10, 2, 7, 3, 4, 5, 3, 10, output="pandoc")
```

Description

Generic functions for prior and posterior distributions

Usage

```

dprior(model, parameter, ...)

pprior(model, parameter, ...)

qprior(model, parameter, ...)

rprior(model, parameter, ...)

sprior(model, parameter, ...)

dpost(model, parameter, ...)

ppost(model, parameter, ...)

qpost(model, parameter, ...)

rpost(model, parameter, ...)

spost(model, parameter, ...)

```

Arguments

model	an object of class <code>brr</code> (see Brr)
parameter	a character string among <code>mu</code> , <code>phi</code> , <code>lambda</code> , <code>x</code> , <code>y</code>
...	the first argument of the function called

Examples

```

model <- Brr(a=2, b=4)
dprior(model, "mu", 1:3)
# the same:
dprior_mu(mu=1:3, a=2, b=4)
## Not run:
dprior(model, "lambda", 1:3)
## End(Not run)
model <- model(c=4, d=5, S=10, T=10)
dprior(model, "lambda", 1:3)
model <- model(x=5, y=10)
ppost(model, "phi", 1)
model <- Brr()
## Not run:
ppost(model, "phi", 1)
## End(Not run)
model <- model(x=5, y=10, S=3, T=10)
ppost(model, "phi", 1)

```

Prior_lambda	<i>Prior distribution on the incidence rate in the treated group</i>
--------------	--

Description

Density, distribution function (see Details) and random generation for the prior distribution on the rate in the treated group. The prior distribution on the incidence rate λ is not to be set by the user: it is induced by the user-specified prior on μ and ϕ .

Usage

```
dprior_lambda(lambda, a, b, c, d, S, T)
```

```
rprior_lambda(n, a, b, c, d, S, T)
```

```
pprior_lambda(q, a, b, c, d, S, T, ...)
```

```
sprior_lambda(a, b, c, d, S, T, ...)
```

Arguments

lambda, q	vector of quantiles
a, b	non-negative shape and rate parameter of the Gamma prior distribution on μ
c, d	non-negative shape parameters of the prior distribution on ϕ
S, T	sample sizes in control group and treated group
n	number of observations to be simulated
...	other arguments passed to genhypergeo through pGB2 , such as <code>series=FALSE</code> to use the continued fraction expansion, or passed to summary_GB2 (for <code>sprior_lambda</code>)

Details

The pdf of the prior distribution on the incidence rate λ involves the Kummer confluent hypergeometric function of the second kind. The cdf involves the generalized hypergeometric function. Its current implementation does not work when $a-c$ is an integer.

Value

`dprior_lambda` gives the density, `pprior_lambda` the distribution function (see Details), `rprior_lambda` samples from the distribution, and `sprior_lambda` gives a summary of the distribution.

Note

`Prior_lambda` is a generic name for the functions documented.

Examples

```
curve(dprior_lambda(x, 2, 2, 2.5, 2, 10, 10), from=0, to=5)
sprior_lambda(2, 2, 2.5, 2, 10, 10)
```

 Prior_mu

Prior distribution on the rate in the control group

Description

Density, distribution function, quantile function and random generation for the prior distribution on the rate in the control group.

Usage

```
dprior_mu(mu, a, b, ...)
pprior_mu(q, a, b, ...)
qprior_mu(p, a, b, ...)
rprior_mu(n, a, b, ...)
sprior_mu(a, b, ...)
```

Arguments

mu, q	vector of quantiles
a, b	non-negative shape parameter and rate parameter
...	other arguments passed to GammaDist or to summary_gamma
p	vector of probabilities
n	number of observations to be simulated

Details

The prior distribution on the rate μ is the Gamma distribution with shape parameter a and rate parameter b

Value

dprior_mu gives the density, pprior_mu the distribution function, qprior_mu the quantile function, and rprior_mu samples from the distribution.

Note

Prior_mu is a generic name for the functions documented.

Examples

```
curve(dprior_mu(x, 2, 2), from=0, to=3)
sprior_mu(2, 2, output="pandoc")
```

 Prior_phi

Prior distribution on the relative risk and the vaccine efficacy

Description

Density, distribution function, quantile function and random generation for the prior distribution on relative risk or the vaccine efficacy.

Usage

```
dprior_phi(phi, b, c, d, S, T, ...)
dprior_VE(VE, b, c, d, S, T, ...)
pprior_phi(q, b, c, d, S, T, ...)
pprior_VE(q, b, c, d, S, T, ...)
qprior_phi(p, b, c, d, S, T, ...)
qprior_VE(p, b, c, d, S, T, ...)
rprior_phi(n, b, c, d, S, T)
sprior_phi(b, c, d, S, T, ...)
```

Arguments

phi, VE, q	vector of quantiles
b	non-negative rate parameter
c, d	non-negative shape parameters
S, T	sample sizes in control group and treated group
...	other arguments passed to Beta2Dist
p	vector of probabilities
n	number of observations to be simulated

Details

The prior distribution on the relative risk ϕ is the Beta2 distribution with shape parameters c and d and scale parameter $(T + b)/S$.

Value

dprior_phi gives the density, pprior_phi the distribution function, qprior_phi the quantile function, rprior_phi samples from the distribution, and sprior_phi gives a summary of the distribution.

Note

Prior_phi is a generic name for the functions documented.

Examples

```
curve(dprior_phi(x, 2, 2, 2, 10, 10), from=0, to=7)
sprior_phi(2, 2, 2, 10, 10, output="pandoc")
```

Prior_x

Prior predictive distribution of the count in the treated group

Description

Density, distribution function, quantile function and random generation for the prior predictive distribution of the count in the treated group.

Usage

```
dprior_x(x, a, b, c, d, T)
```

```
pprior_x(q, a, b, c, d, T)
```

```
qprior_x(p, a, b, c, d, T)
```

```
rprior_x(n, a, b, c, d, T)
```

```
sprior_x(a, b, c, d, T, ...)
```

Arguments

x,q	vector of non-negative integer quantiles
a,b	non-negative shape parameter and rate parameter of the Gamma prior distribution on the rate μ
c,d	non-negative shape parameters of the prior distribution on ϕ
T	sample size of the control group
p	vector of probabilities
n	number of observations to be simulated
...	passed to summary_PGB2

Details

The prior predictive distribution of the count x is the [Poisson-Gamma-Beta2 distribution](#) with shape parameters a , d , c , and hyperrate parameter $b/(b + T)$.

Value

`dprior_x` gives the density, `pprior_x` the distribution function, `qprior_x` the quantile function, `rprior_x` samples from the distribution, and `sprior_x` gives a summary of the distribution.

Note

`Prior_x` is a generic name for the functions documented.

Examples

```
barplot(dprior_x(0:30, 2, 3, 4, 5, 10))
sprior_x(2, 3, 4, 5, 10, output="pandoc")
```

<code>Prior_x_given_y</code>	<i>Prior predictive distribution of the count x in the treated group conditionally to the count y in the treated group</i>
------------------------------	--

Description

Density, distribution function, quantile function and random generation for the conditional prior predictive distribution of x given y .

Usage

```
dprior_x_given_y(x, y, a, c, d)
pprior_x_given_y(q, y, a, c, d)
qprior_x_given_y(p, y, a, c, d)
rprior_x_given_y(n, y, a, c, d)
sprior_x_given_y(y, a, c, d, ...)
```

Arguments

<code>x, q</code>	vector of non-negative integer quantiles
<code>y</code>	count (integer) in the control group
<code>a</code>	non-negative shape parameter of the Gamma prior distribution on the rate μ
<code>c, d</code>	non-negative shape parameters of the prior distribution on ϕ
<code>p</code>	vector of probabilities
<code>n</code>	number of observations to be simulated
<code>...</code>	arguments passed to summary_beta_nbinom

Details

The prior predictive distribution of the count x is the [Beta-negative binomial distribution](#) with shape parameters $a + y$, d , c .

Value

`dprior_x_given_y` gives the density, `pprior_x_given_y` the distribution function, `qprior_x_given_y` the quantile function, `rprior_x_given_y` samples from the distribution, and `sprior_x_given_y` gives a summary of the distribution.

Note

`Prior_x_given_y` is a generic name for the functions documented.

Examples

```
barplot(dprior_x_given_y(0:10, 5, 3, 10, 20))
sprior_x_given_y(5, 3, 10, 20, output="pandoc")
```

<code>Prior_y</code>	<i>Prior predictive distribution of the count in the control group</i>
----------------------	--

Description

Density, distribution function, quantile function and random generation for the prior predictive distribution of the count in the control group.

Usage

```
dprior_y(y, a, b, T, ...)
```

```
pprior_y(q, a, b, T, ...)
```

```
qprior_y(p, a, b, T, ...)
```

```
rprior_y(n, a, b, T)
```

```
sprior_y(a, b, T, ...)
```

Arguments

<code>y, q</code>	vector of non-negative integer quantiles
<code>a, b</code>	non-negative shape parameter and rate parameter of the Gamma prior distribution on the rate μ
<code>T</code>	sample size of the control group
<code>...</code>	other arguments passed to NegBinomial or summary_nbinom (for <code>sprior_y</code>)
<code>p</code>	vector of probabilities
<code>n</code>	number of observations to be simulated

Details

The prior predictive distribution of the count y is the Poisson-Gamma distribution with shape parameter a and hyperrate parameter b/T , which is also the negative binomial distribution with shape a and probability of success $b/(b + T)$.

Value

`dprior_y` gives the density, `pprior_y` the distribution function, `qprior_y` the quantile function, `rprior_y` samples from the distribution, and `sprior_y` gives a summary of the distribution.

Note

`Prior_y` is a generic name for the functions documented.

Examples

```
barplot(dprior_y(0:10, 2, 2, 1))
sprior_y(2, 2, 1, output="pandoc")
```

summary_gamma

Summary of a Gamma distribution

Description

Mode, mean, variance, and quartiles for a Gamma distribution with shape parameter a and rate parameter b .

Usage

```
summary_gamma(a, b, output = "list", ...)
```

Arguments

<code>a, b</code>	Shape and rate parameters.
<code>output</code>	"list" to return a list, "pandoc" to print a table
<code>...</code>	arguments passed to pander.data.frame

Examples

```
summary_gamma(a=2, b=4, output="pandoc", style="rmarkdown")
```

summary_nbinom	<i>Summary of a Negative Binomial distribution</i>
----------------	--

Description

Mode, mean, variance, and quartiles for a Negative Binomial distribution with shape parameter size and probability parameter prob.

Usage

```
summary_nbinom(size, prob, output = "list", ...)
```

Arguments

size, prob	parameters of the negative binomial distribution (as for NegBinomial)
output	"list" to return a list, "pandoc" to print a table
...	arguments passed to pander.data.frame

Examples

```
summary_nbinom(size=2, prob=0.4, output="pandoc", style="rmarkdown")
```

Index

- * **bayesian**
 - brr-package, 2
 - * **distribution**
 - brr-package, 2
 - * **htest**
 - brr-package, 2
 - * **models**
 - brr-package, 2
 - * **package**
 - brr-package, 2
- barplot, 18
- Beta-negative binomial distribution, 16
- Beta2 distribution, 8, 16
- Beta2Dist, 3, 21, 28
- BNBDist, 4
- Brr, 5, 18, 25
- brr, 11
- brr (brr-package), 2
- brr-package, 2
- brr_estimates, 11
- brr_estimates (Inference), 10
- brr_intervals, 11
- brr_intervals (Inference), 10
- coef.brr (inference.brr), 11
- confint.brr, 10
- confint.brr (inference.brr), 11
- dbeta2 (Beta2Dist), 3
- dbeta_nbinom (BNBDist), 4
- dGB2 (GB2Dist), 7
- dGIB (GIBDist), 9
- dPGB2 (PGB2Dist), 16
- dPGIB (PGIBDist), 17
- dpost (PriorAndPosterior), 24
- dpost_lambda (Posterior_lambda), 19
- dpost_mu (Posterior_mu), 20
- dpost_phi (Posterior_phi), 21
- dpost_VE (Posterior_phi), 21
- dpost_x (Post_x), 22
- dpost_y (Post_y), 23
- dprior (PriorAndPosterior), 24
- dprior_lambda (Prior_lambda), 26
- dprior_mu (Prior_mu), 27
- dprior_phi (Prior_phi), 28
- dprior_VE (Prior_phi), 28
- dprior_x (Prior_x), 29
- dprior_x_given_y (Prior_x_given_y), 30
- dprior_y (Prior_y), 31
- FDist, 3
- FrequentistInference, 6
- Gamma-Inverse Beta distribution, 17
- GammaDist, 27
- GB2Dist, 7
- genhypergeo, 8, 26
- ghyper, 5
- GIBDist, 9, 19, 20
- Inference, 10
- inference.brr, 11
- integrate, 13, 14
- intrinsic2_bounds
 - (Intrinsic2Inference), 12
- intrinsic2_discrepancy, 13
- intrinsic2_estimate
 - (Intrinsic2Inference), 12
- intrinsic2_H0 (Intrinsic2Inference), 12
- intrinsic2_phi0 (Intrinsic2Inference), 12
- intrinsic2_phi0_sims
 - (Intrinsic2Inference), 12
- Intrinsic2Inference, 10, 12
- intrinsic_bounds (IntrinsicInference), 14
- intrinsic_discrepancy, 15
- intrinsic_estimate
 - (IntrinsicInference), 14

- intrinsic_H0 (IntrinsicInference), 14
- intrinsic_phi0 (IntrinsicInference), 14
- intrinsic_phi0_sims
(IntrinsicInference), 14
- IntrinsicInference, 10, 14
- moment_GB2 (GB2Dist), 7
- NegBinomial, 31, 33
- pander.data.frame, 9, 17, 32, 33
- pandoc.table.return, 6, 11
- pbeta2 (Beta2Dist), 3
- pbeta_nbinom (BNBDist), 4
- pGB2, 26
- pGB2 (GB2Dist), 7
- PGB2Dist, 16
- PGIBDist, 17
- plot, 18
- plot.brr, 18
- Post_x, 22
- Post_y, 23
- Posterior_lambda, 19
- Posterior_mu, 20
- Posterior_phi, 21
- pPGB2 (PGB2Dist), 16
- pPGIB (PGIBDist), 17
- ppost (PriorAndPosterior), 24
- ppost_phi (Posterior_phi), 21
- ppost_VE (Posterior_phi), 21
- ppost_x (Post_x), 22
- ppost_y (Post_y), 23
- pprior (PriorAndPosterior), 24
- pprior_lambda (Prior_lambda), 26
- pprior_mu (Prior_mu), 27
- pprior_phi (Prior_phi), 28
- pprior_VE (Prior_phi), 28
- pprior_x (Prior_x), 29
- pprior_x_given_y (Prior_x_given_y), 30
- pprior_y (Prior_y), 31
- predict.brr (inference.brr), 11
- print.coef.brr (inference.brr), 11
- print.confint.brr (inference.brr), 11
- print.predict.brr (inference.brr), 11
- print.summary.brr (Brr), 5
- Prior_lambda, 26
- Prior_mu, 27
- Prior_phi, 28
- Prior_x, 29
- Prior_x_given_y, 30
- Prior_y, 31
- PriorAndPosterior, 24
- qbeta2 (Beta2Dist), 3
- qbeta_nbinom (BNBDist), 4
- qGB2 (GB2Dist), 7
- qPGB2 (PGB2Dist), 16
- qPGIB (PGIBDist), 17
- qpost (PriorAndPosterior), 24
- qpost_phi (Posterior_phi), 21
- qpost_VE (Posterior_phi), 21
- qpost_x (Post_x), 22
- qpost_y (Post_y), 23
- qprior (PriorAndPosterior), 24
- qprior_mu (Prior_mu), 27
- qprior_phi (Prior_phi), 28
- qprior_VE (Prior_phi), 28
- qprior_x (Prior_x), 29
- qprior_x_given_y (Prior_x_given_y), 30
- qprior_y (Prior_y), 31
- rbeta2 (Beta2Dist), 3
- rbeta_nbinom (BNBDist), 4
- rGB2 (GB2Dist), 7
- rGIB (GIBDist), 9
- rPGB2 (PGB2Dist), 16
- rPGIB (PGIBDist), 17
- rpost (PriorAndPosterior), 24
- rpost_lambda (Posterior_lambda), 19
- rpost_mu (Posterior_mu), 20
- rpost_phi (Posterior_phi), 21
- rpost_x (Post_x), 22
- rpost_y (Post_y), 23
- rprior (PriorAndPosterior), 24
- rprior_lambda (Prior_lambda), 26
- rprior_mu (Prior_mu), 27
- rprior_phi (Prior_phi), 28
- rprior_x (Prior_x), 29
- rprior_x_given_y (Prior_x_given_y), 30
- rprior_y (Prior_y), 31
- rr_interval_binomial
(FrequentistInference), 6
- rr_interval_SK (FrequentistInference), 6
- rr_intervals (FrequentistInference), 6
- sbeta_nbinom (BNBDist), 4
- spost (PriorAndPosterior), 24
- spost_lambda (Posterior_lambda), 19

spost_mu (Posterior_mu), 20
spost_phi (Posterior_phi), 21
spost_x (Post_x), 22
spost_y (Post_y), 23
sprior (PriorAndPosterior), 24
sprior_lambda (Prior_lambda), 26
sprior_mu (Prior_mu), 27
sprior_phi (Prior_phi), 28
sprior_x (Prior_x), 29
sprior_x_given_y (Prior_x_given_y), 30
sprior_y (Prior_y), 31
summary.brr (Brr), 5
summary_beta2 (Beta2Dist), 3
summary_beta_nbinom, 30
summary_beta_nbinom (BNBDist), 4
summary_gamma, 27, 32
summary_GB2, 26
summary_GB2 (GB2Dist), 7
summary_GIB (GIBDist), 9
summary_nbinom, 31, 33
summary_PGB2, 29
summary_PGB2 (PGB2Dist), 16
summary_PGIB, 23, 24
summary_PGIB (PGIBDist), 17