Package ‘brr’

August 29, 2016

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
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](https://github.com/stla/brr).

Details

- Package: `brr`
- Type: Package
- Version: 1.0.0
- Date: 2015-09-07
- License: GPL-2
Beta2Dist

Author(s)
Stéphane Laurent

References

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
x, q vector of quantiles
c, d non-negative shape parameters
scale non-negative scale parameter
log logical; if true, returns the logarithm of the result
... other arguments passed to FDist
p vector of probabilities
n number of observations to be simulated
output type of the summary_beta2 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 \( k \ast (U/(1 - U)) \) 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

```r
curve(dbeta2(x, 3, 10, scale=2), from=0, to=3)
u <- rbeta(1e5, 3, 10)
lines(density(2*x/(1-x)), 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

- `x,q`: vector of non-negative integer quantities
- `a,c,d`: non-negative shape parameters
- `...`: other arguments passed to `ghyper`
- `p`: vector of probabilities
- `n`: number of observations to be sampled
- `output`: type of the `summary_beta_nbinom` 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

```r
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**

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

## S3 method for class 'brr'

```r
summary(object, phi0 = 1, hypothesis = "greater", ...)
```

## S3 method for class 'summary.brr'

```r
print(x, table.style = "grid", ...)
```
FrequentistInference

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 \( P_r(\phi > \phi_0) \), "lower" to return \( P_r(\phi < \phi_0) \)

x the output to be printed

table.style the style of the table to print (passed to \texttt{pandoc.table.return})

Value

\texttt{Brr} returns an object of class \texttt{brr}, \texttt{summary.brr} returns a list but prints its contents through \texttt{print.summary.brr}

Examples

```r
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)
```

---

\textbf{FrequentistInference}  \textit{Frequentist inference about the relative risk}

Description

Frequentist confidence intervals about the relative risk: \texttt{binomial interval (rr_interval_binomial)} and Sahai and Khurshid confidence interval \texttt{(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


Examples

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

GB2Dist

Description

Density and random generation for the Gamma-Beta2 distribution with shape parameters `a`, `c`, `d` and rate parameter `tau` (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

x, q vector of non-negative quantiles
a, c, d non-negative shape parameters
tau non-negative rate parameter
... arguments passed to genhypergeo function
p vector of probabilities
n number of observations to be sampled
k the order of the moment
output type of the summary_gb2 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 \( \tau \) 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

```r
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)
```
GIBDist

Gamma-Inverse Beta distribution

Description

Density and random generation for the Gamma-Inverse Beta distribution with shape parameters a, alpha, beta and rate parameter rho.

Usage

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

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

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

Arguments

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

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")
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

```r
## 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

```r
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)
```

Description

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

Usage

```r
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

- `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
... arguments passed to \texttt{integrate}
nims number of simulations
otol desired accuracy for optimization
phi.star the hypothesized value of \phi
alternative alternative hypothesis; "less" for H1: \phi 0 < \text{phi.star}, "greater" for H1: \phi 0 > \text{phi.star}
conf credibility level
parameter parameter of interest: relative risk "phi" or vaccine efficacy "VE"

\textbf{Value}

\texttt{intrinsic2\_phi0} returns the posterior expected loss, \texttt{intrinsic2\_estimate} returns the intrinsic estimate, \texttt{intrinsic2\_H0} performs intrinsic hypothesis testing, and \texttt{intrinsic2\_bounds} returns the intrinsic credibility interval.

\textbf{Examples}

\begin{verbatim}
a<-2; b<-10; c<-1/2; d<-0; s<-100; T<-5; 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)
\end{verbatim}

\texttt{intrinsic2\_discrepancy}

\textit{Second intrinsic discrepancy}

\textbf{Description}

Intrinsic discrepancy from \phi 0 to \phi in the marginal model.

\textbf{Usage}

\texttt{intrinsic2\_discrepancy(\phi 0, \phi, a, b, S, T)}

\textbf{Arguments}

\begin{itemize}
  \item \phi 0 the proxy value of \phi
  \item \phi the true value of the parameter
  \item a, b, the parameters of the prior Gamma distribution on \mu
  \item S, T sample sizes
\end{itemize}

\textbf{Value}

A number, the intrinsic discrepancy from \phi 0 to \phi.
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 H1: phi0 < phi.star, "greater" for H1: phi0 > 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

```r
a <- PNU[b <- P[c <- 1/R[d <- P[s <- 1PP[t <- s[x <- P[y <- RP

intrinsic_phiPHPNUL xL yL sL tL aL bL cL dI

intrinsic_estimateHxL yL sL tL aL bL cL dI

bounds <- intrinsic_boundsHxL yL sL tL aL bL cL dL conf=0.95); bounds

ppost_phiHbounds[2], a, b, c, d, S, T, x, yI- ppost_phiHbounds[1], a, b, c, d, S, T, x, y)
```

---

**intrinsic_discrepancy**  
Intrinsic discrepancy

**Description**

Intrinsic discrepancy from \( \phi_0 \) to \((\mu, \phi)\).

**Usage**

```r
intrinsic_discrepancy(\phi_0, \mu, \phi, S, T)
```

**Arguments**

- \( \phi_0 \): the proxy value of \( \phi \)
- \( \mu, \phi \): the true values of the parameters
- \( S, T \): sample sizes

**Value**

A number, the intrinsic discrepancy from \( \phi_0 \) to \((\mu, \phi)\).
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 \( \tau \) (scale of the Beta2 distribution). For \( \tau=1 \) this is the same as the Beta-negative binomial distribution.

Usage
\[
\begin{align*}
dPGB2(x, a, c, d, \tau) \\
pPGB2(q, a, c, d, \tau) \\
qPGB2(p, a, c, d, \tau) \\
rPGB2(n, a, c, d, \tau) \\
\text{summary_PGB2}(a, c, d, \tau, \text{output} = \text{"list"})
\end{align*}
\]

Arguments
\[
\begin{align*}
x, q & \quad \text{vector of non-negative integer quantiles} \\
a, c, d & \quad \text{non-negative shape parameters} \\
\tau & \quad \text{non-negative hyperrate parameter} \\
p & \quad \text{vector of probabilities} \\
n & \quad \text{number of observations to be sampled} \\
\text{output} & \quad \text{type of the \text{summary_PGB2} output: "list" to return a list, "pandoc" to print a table}
\end{align*}
\]

Details
This is the mixture distribution obtained by sampling a value \( y \) from the Beta2 distribution with shape parameters \( c, d \), and scale \( \tau \), then sampling a value \( \lambda \) from the Gamma distribution with shape \( a \) and rate \( y \), and then sampling the Poisson distribution with mean \( \lambda \).

Value
\( dPGB2 \) gives the density, \( pPGB2 \) the cumulative function, \( rPGB2 \) samples from the distribution, and \( \text{summary_PGB2} \) gives a summary of the distribution.

Note
PGB2Dist is a generic name for the functions documented.
Examples

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

---

**PGBDist**  
*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 $\rho$.

**Usage**

```r
dPGB(x, a, alpha, beta, rho)
pPGB(q, a, alpha, beta, rho)
qPGB(p, a, alpha, beta, rho)
rPGB(n, a, alpha, beta, rho)
summary_PGB(a, alpha, beta, rho, output = "list", ...)
```

**Arguments**

- `x, q` vector of **integer** quantiles
- `a` non-negative shape parameter of the Gamma distribution
- `alpha, beta` non-negative shape parameters of the mixing Beta distribution
- `rho` hyperrate parameter (rate of the mixing distribution)
- `p` vector of probabilities
- `n` number of observations to be simulated
- `output` type of the `summary_PGB` 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**

`dPGB` gives the density, `rPGB` samples from the distribution, and `summary_PGB` gives a summary of the distribution.
Note
PGIBDist is a generic name for the functions documented.

Examples

```r
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

### Description

plot brr

### Usage

```r
## 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 `dprior(mu)` 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

```r
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, aprior(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 distribution on the incidence rate in the treated group

**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 $\mu$
- `c, d` non-negative shape parameters of the prior distribution on $\phi$
- `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 $\lambda$ 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 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 \( \mu \)
c,d non-negative shape parameters of the prior distribution on \( \phi \)
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 \( \mu \) 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
- `
- `n`: number of observations to be simulated
- `p`: vector of probabilities

**Details**

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

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

Posterior_x 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_x(a, b, c, d, S, T, x, y, output="pandoc")
require(magrittr)
phi <- seq(0, 6, length.out=100)
phi %>% { plot(., dpost_x(., a, b, c, d, S, T, x, y), type="l") }
phi %>% { lines(., dprior_x(., a, b, c, d, T), col="red") }

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 \( \mu \)
c, d non-negative shape parameters of the prior distribution on \( \phi \)
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
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

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

- \( y_{new}, q \): vector of non-negative integer quantiles
- \( a, b \): non-negative shape parameter and rate parameter of the Gamma prior distribution on the rate \( \mu \)
- \( c, d \): non-negative shape parameters of the prior distribution on \( \phi \)
- \( x, y \): counts (integer) in the treated group and control group of the observed experiment
- \( T, T_{new} \): 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

```r
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")
```
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 brr (see Brr)
parameter a character string among mu, phi, lambda, x, y

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 distribution on the incidence rate in the treated group

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 \( \lambda \) is not to be set by the user: it is induced by the user-specified prior on \( \mu \) and \( \phi \).

Usage

\[
\begin{align*}
\text{dprior\_lambda}(\text{lambda, a, b, c, d, S, T}) \\
r\text{prior\_lambda}(n, a, b, c, d, S, T) \\
p\text{prior\_lambda}(q, a, b, c, d, S, T, \ldots) \\
s\text{prior\_lambda}(a, b, c, d, S, T, \ldots)
\end{align*}
\]

Arguments

- \text{lambda,q} : vector of quantiles
- \text{a,b} : non-negative shape and rate parameter of the Gamma prior distribution on \( \mu \)
- \text{c,d} : non-negative shape parameters of the prior distribution on \( \phi \)
- \text{S,T} : sample sizes in control group and treated group
- \text{n} : number of observations to be simulated
- \ldots : other arguments passed to \text{genhypergeo} through \text{pgB2}, such as series=FALSE to use the continued fraction expansion, or passed to \text{summary\_GB2} (for \text{sprior\_lambda})

Details

The pdf of the prior distribution on the incidence rate \( \lambda \) 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

- \text{dprior\_lambda} gives the density,
- \text{pprior\_lambda} the distribution function (see Details),
- \text{rprior\_lambda} samples from the distribution, and
- \text{sprior\_lambda} gives a summary of the distribution.

Note

\text{Prior\_lambda} is a generic name for the functions documented.
Prior_mu

Examples

```r
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 $\mu$ 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,
- `rprior_mu` samples from the distribution.

Note

`Prior_mu` is a generic name for the functions documented.
Prior_phi

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 \( \phi \) is the Beta2 distribution with shape parameters \( c \) and \( d \) and scale parameter \( (T + b)/S \).
Prior_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

```r
curve(dprior_x(x, 2, 2, 2, 10, 10), from=0, to=7)
sprior_x(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

```r
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 $\mu$
- `c, d`: non-negative shape parameters of the prior distribution on $\phi$
- `T`: sample size of the control group
- `p`: vector of probabilities
- `n`: number of observations to be simulated
- `...`: passed to `summary_PGB2`
Prior_x_given_y

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

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

Prior_x_given_y

Prior predictive distribution of the count \( x \) in the treated group conditionally to the count \( y \) in the treated group

Description

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

Usage

```r
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

- \( x, q \) vector of non-negative integer quantiles
- \( y \) count (integer) in the control group
- \( a \) non-negative shape parameter of the Gamma prior distribution on the rate \( \mu \)
- \( c, d \) non-negative shape parameters of the prior distribution on \( \phi \)
- \( p \) vector of probabilities
- \( n \) number of observations to be simulated
- \( ... \) 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

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

<table>
<thead>
<tr>
<th>Prior_y</th>
<th>Prior predictive distribution of the count in the control group</th>
</tr>
</thead>
</table>

Description

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

Usage

```r
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

- \( y, q \) vector of non-negative \textbf{integer} quantiles
- \( a, b \) non-negative shape parameter and rate parameter of the Gamma prior distribution on the rate \( \mu \)
- \( T \) sample size of the control group
- \( ... \) other arguments passed to \texttt{NegBinomial} or \texttt{summary_nbinom} (for sprior_y)
- \( p \) vector of probabilities
- \( n \) 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

```r
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**

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

**Arguments**

- `a, b` Shape and rate parameters.
- `output` "list" to return a list, "pandoc" to print a table
- `...` arguments passed to `pander.data.frame`

**Examples**

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

Summary of a Negative Binomial distribution

---

### Description

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

### Usage

```r
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

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

*Topic **bayesian**
  brr-package, 2
*Topic **distribution**
  brr-package, 2
*Topic **htest**
  brr-package, 2
*Topic **models**
  brr-package, 2
*Topic **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

coeff.brr (inference.brr), 11
dconfint.brr, 10
dconfint.brr (inference.brr), 11

dbeta2 (Beta2Dist), 3
dbeta_nbinom (BNBDist), 4
dGB2 (GB2Dist), 7
dIB (GBDist), 9
dPGB2 (PGB2Dist), 16
dPGBI (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
gyper, 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 (BetaDist), 3
pbeta_nbinom (BNBDist), 4
pGB2, 26
pGB2 (GB2Dist), 7
PG2Dist, 16
PGIBDist, 17
plot, 18
plot.brr, 18
Post_x, 22
Post_y, 23
Posterior_lambda, 19
Posterior_mu, 20
Posterior_phi, 21
PGB2 (PG2Dist), 16
PGIB (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 coef.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_y, 30
PriorAndPosterior, 24
qbeta2 (Beta2Dist), 3
qbeta_nbinom (BNBDist), 4
qGB2 (GB2Dist), 7
qGB2 (PG2Dist), 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
rGB2 (PG2Dist), 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
qprior Lambda (Prior_lambda), 26
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
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_PGB (PGIBDist), 23, 24
summary_PGIB (PGIBDist), 17