Package 'Rgbp'

August 14, 2013

Version 1.0.1

Date 2013-03-16

Title Gaussian, Poisson, and Binomial Hierarchical Modeling

Author Joseph Kelly, Hyungsuk Tak, and Carl Morris

Maintainer Joseph Kelly <kelly2@fas.harvard.edu>

Depends sn

Suggests mnormt

Description

Rgbp is an R package that utilizes approximate Bayesian machinery to provide a method of estimating two-level hierarchical models for Gaussian, Poisson, and Binomial data in a fast and computationally efficient manner. The main products of this package are point and interval estimates for the true parameters, whose good frequency properties can be validated via its repeated sampling procedure called frequency method checking. It is found that such Bayesian-frequentist reconciliation allows Rgbp to have attributes desirable both sides of the aisle, working well in small samples and yielding good coverage probabilities for its interval estimates.

License GPL-2

BugReports https://github.com/jyklly/Rgbp/issues

R topics documented:

---

baseball *Baseball Data*

Description

Batting averages of 18 major league players through their first 45 official at bats of the 1970 season. These batting averages were published weekly in the New York Times, and by April 26, 1970.

Usage

```
data(baseball)
```

Format

A data set of 18 players with 12 covariates:

FirstName each player's first name

LastName each player's last name

At.Bats number of times batted

Hits each player's number of hits among 45 at bats

BattingAverage batting averages among 45 at bats

RemainingAt.Bats number of times batted after 45 at bats until the end of season

RemainingAverage batting averages after 45 at bats until the end of season

SeasonAt.Bats number of times batted over the whole season

SeasonHits each player's number of hits over the whole season

SeasonAverage batting averages over the whole season

League 1 if a player is in the National league

Position each player's position

Source

Efron, B. and Morris, C. (1975). Data Analysis Using Stein's Estimator and its Generalizations. *Journal of the American Statistical Association*. **70**. 311-319.

Examples

```
data(baseball)
```

---

coverage   *Estimating Coverage Probability*

Description

coverage estimates Rao-Blackwellized and simple unbiased coverage probabilities.

Usage

```
coverage(gbp.object, A.or.r, reg.coef, mean.PriorDist, nsim = 10)
```

Arguments

gbp.object   a resultant object of gbp function.

A.or.r   (optional) a given true numeric value of *A* for Gaussian data or of *r* for Binomial
                   and Poisson data. If not designated, the estimated value in the gbp.object
                   object will be considered as a true value.

reg.coef   (optional) a given true (*m* by 1) vector for regression coefficients, β, where *m* is
                   the number of regression coefficients including an intercept. If not designated,
                   the estimated value in the gbp.object object will be considered as a true value.

mean.PriorDist   (optional) a given true numeric value for the mean of (second-level) prior distri-
                   bution. If not designated, the previously known value in the gbp.object object
                   will be considered as a known prior mean.

nsim   number of datasets to be generated. Default is 10.

Details

As for the argument gbp.object, if the result of gbp is designated to b, for example
"b <- gbp(z, n, model = "binomial")", the argument gbp.object indicates this b.

Data generating process is based on a second-level hierarchical model. The first-level hierarchy is a
distribution of observed data and the second-level is a conjugate prior distribution on the first-level
parameter.

To be specific, for Normal data, gbp constructs a two-level Normal-Normal 2-level model. σ_j^2 below
is assumed to be known or to be accurately estimated (s_j^2) and subscript *j* indicates *j*-th group in a
dataset.

$$(y_j\tilde{~}|\tilde{~}\theta_j)\tilde{~} \sim \tilde{~}indep\tilde{~}N(\theta_j, \sigma_j^2)$$

$$(\theta_j\tilde{~}|\tilde{~}\mu_{0j}, \tilde{~}A)\tilde{~} \sim \tilde{~}indep\tilde{~}N(\mu_{0j}, \tilde{~}A)$$

$$\mu_{0j}\tilde{~} = \tilde{~}x_j'\beta$$

for $j = 1, \ldots, k$, where *k* is the number of groups (units) in a dataset.

For Poisson data, gbp builds a two-level Poisson-Gamma multi-level model. A square bracket be-
low indicates [mean, variance] of distribution and a constant multiplied to the notation representing

Gamma distribution (Gam) is a scale. Also, for consistent notation, $y_j = \frac{z_j}{n_j}$ and n_j can be interpreted as *j*-th group's exposure only in this Poisson-Gamma hierarchical model.

$$(z_j{}^\sim|{}^\sim\theta_j){}^\sim \sim {}^\sim indep{}^\sim Pois(n_j\theta_j)$$

$$(\theta_j{}^\sim|{}^\sim r, {}^\sim\mu_{0j}){}^\sim \sim {}^\sim indep{}^{\sim\sim}\frac{1}{r}Gam(r\mu_{0j}){}^\sim \sim {}^\sim indep{}^\sim Gam[\mu_{0j}, \mu_{0j}/r]$$

$$log(\mu_{0j}){}^\sim = {}^\sim x'_j\beta$$

for $j = 1, \ldots, k$, where *k* is the number of groups (units) in a dataset.

For Binomial data, gbp sets a two-level Binomial-Beta multi-level model. For reference, a square bracket below indicates [mean, variance] of distribution and $y_j = \frac{z_j}{n_j}$.

$$(z_j{}^\sim|{}^\sim\theta_j){}^\sim \sim {}^\sim indep{}^\sim Bin(n_j, \theta_j)$$

$$(\theta_j{}^\sim|{}^\sim r, \mu_{0j}){}^\sim \sim {}^\sim indep{}^\sim Beta(r\mu_{0j}, {}^\sim r(1-\mu_{0j})){}^\sim \sim {}^\sim indep{}^\sim Beta[\mu_{0j}, {}^\sim\mu_{0j}(1-\mu_{0j}){}^\sim/{}^\sim(r+1)]$$

$$logit(\mu_{0j}){}^\sim = {}^\sim x'_j\beta$$

for $j = 1, \ldots, k$, where *k* is the number of groups (units) in a dataset.

From now on, the subscript *(i)* means *i*-th simulation and the subscript *j* indicates *j*-th group. So, notations with a subscript *(i)* are (*k* by 1) vectors, for example $\theta'_{(i)} = (\theta_{(i)1}, \theta_{(i)2}, \ldots, \theta_{(i)k})$.

Pseudo-data generating process starts from the second-level hierarchy to the first-level. coverage first generates true parameters ($\theta_{(i)}$) for *k* groups at the second-level and then moves onto the first-level to simulate pseudo-data sets, $y_{(i)}$ for Gaussian or $z_{(i)}$ for Binomial and Poisson data, given previously generated true parameters ($\theta_{(i)}$).

So, in order to generate pseudo-datasets, coverage needs parameters of prior distribution, (*A* (or *r*) and β (reg.coef)) or (*A* (or *r*) and μ_0). From here, we have four options to run coverage.

First, if any values related to the prior distribution are not designated like coverage(b, nsim = 10), then coverage will regard estimated values (or known prior mean, μ_0) in b (gbp.object) as given true values when it generates lots of pseudo-datasets. After sampling $\theta_{(i)}$ from the prior distribution determined by these estimated values (or known prior mean) in b (gbp.object), coverage creates an *i*-th pseudo-dataset based on $\theta_{(i)}$ just sampled.

Second, coverage allows us to try different true values in generating datasets. Suppose gbp.object is based on the model with a known prior mean, μ_0. Then, we can try either different A.or.r or mean.PriorDist. For example, coverage(b, A.or.r = 20, nsim = 10), coverage(b, mean.PriorDist = 0.5, nsim or coverage(b, A.or.r = 20, mean.PriorDist = 0.5, nsim = 10). Note that we cannot set reg.coef because the second-level mean (prior mean) is known in gbp.object to begin with.

Suppose gbp.object is based on the model with an unknown prior mean. In this case, gbp.object has the estimation result of regression model, linear regression for Normal-Normal, log-linear regression for Poisson-Gamma, or logistic regression for Binomial-Beta, (only intercept term if there is no covariate) to estimate the unknown prior mean. Then, we can try some options: one or two of (A.or.r, mean.PriorDist, reg.coef). For example, coverage(b, A.or.r = 20, nsim = 10), coverage(b, mean.PriorDist = 0.5, nsim = 10), or coverage(b, reg.coef = 0.1, nsim = 10) with no covariate where reg.coef is a designated intercept term. Estimates in gbp.object will be used for undesignated values. Also, we can try appropriate combinations of two arguments. For example, coverage(b, A.or.r = 20, mean.PriorDist = 0.5, nsim = 10) and coverage(b, A.or.r = 20, reg.coef If we have one covariate, a 2 by 1 vector should be designated for reg.coef, one for an intercept term and the other for a regression coefficient of the covariate. Note that the two arguments,

mean.PriorDist and reg.coef, cannot be assigned together because we do not need reg.coef given mean.PriorDist.

The simple unbiased estimator of coverage probability in *j*-th group is a sample mean of indicators over all simulated datasets. The *j*-th indicator in *i*-th simulation is 1 if the estimated interval of the *j*-th group on *i*-th simulated dataset contains a true parameter $\theta_{(i)j}$ that generated the observed value of the *j*-th group in the *i*-th dataset.

Rao-Blackwellized unbiased estimator for group *j* is a conditional expectation of the simple unbiased estimator given a sufficient statistic, y_j for Gaussian or z_j for Binomial and Poisson data.

Value

coverageRB	Rao-Blackwellized unbiased estimate of coverage probability for each group averaged over all simulations.
coverageS	Simple unbiased estimate of coverage probability for each group averaged over all simulations.
average.coverageRB	
	average Rao-Blackwellized unbiased estimate of coverage probability over all the simulation regardless of group.
average.coverageS	
	average simple unbiased estimate of coverage probability over all the simulation regardless of group.
raw.resultRB	all the Rao-Blackwellized unbiased estimates for coverage probabilities for every group and for every simulation.
raw.resultS	all the simple unbiased estimates for coverage probabilities for every group and for every simulation.

Author(s)

Joseph Kelly, Hyungsuk Tak, and Carl Morris

References

Christiansen, C. and Morris, C. (1997). Hierarchical Poisson Regression Modeling. *Journal of the American Statistical Association*. **92**. 618-632.

Examples

```
# Loading datasets
data(schools)
y <- schools$y
se <- schools$se

# Arbitrary covariate for schools data
x2 <- rep(c(-1, 0, 1, 2), 2)

# baseball data where z is Hits and n is AtBats
z <- c(18, 17, 16, 15, 14, 14, 13, 12, 11, 11, 10, 10, 10, 10, 10,  9,  8,  7)
n <- c(45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45)
```

```
# One covariate: 1 if a player is an outfielder and 0 otherwise
x1 <- c(1,  1,  1,  1,  1,  0,  0,  0,  0,  1,  0,  0,  0,  1,  1,  0,  0,  0)

##################################################################
# Gaussian Regression Interactive Multi-level Modeling (GRIMM) #
##################################################################

  ##############################################################################
  # If we do not have any covariate and do not know a mean of the prior distribution #
  ##############################################################################

  g <- gbp(y, se, model = "gaussian")

  ### when we want to simulate pseudo datasets considering the estimated values
  ### as true ones.
  gcv <- coverage(g, nsim = 10)

  ### gcv$coverageRB, gcv$coverageS, gcv$average.coverageRB, gcv$average.coverageS,
  ### gcv$minimum.coverageRB, gcv$raw.resultRB, gcv$raw.resultS

  ### gcv <- coverage(g, mean.PriorDist = 3, nsim = 100)
  ### gcv <- coverage(g, A.or.r = 150, nsim = 100)
  ### gcv <- coverage(g, reg.coef = 10, nsim = 100)
  ### gcv <- coverage(g, A.or.r = 150, mean.PriorDist = 3, nsim = 100)
  ### gcv <- coverage(g, A.or.r = 150, reg.coef = 10, nsim = 100)

  ##############################################################################
  # If we have one covariate and do not know a mean of the prior distribution yet, #
  ##############################################################################

  g <- gbp(y, se, x2, model = "gaussian")

  ### when we want to simulate pseudo datasets considering the estimated values
  ### as true ones.
  gcv <- coverage(g, nsim = 10)

  ### gcv$coverageRB, gcv$coverageS, gcv$average.coverageRB, gcv$average.coverageS,
  ### gcv$minimum.coverageRB, gcv$raw.resultRB, gcv$raw.resultS

  ### gcv <- coverage(g, mean.PriorDist = 3, nsim = 100)
  ### gcv <- coverage(g, A.or.r = 200, nsim = 100)
  ### gcv <- coverage(g, reg.coef = c(10, 2), nsim = 100)
  ### gcv <- coverage(g, A.or.r = 200, mean.PriorDist = 3, nsim = 100)
  ### gcv <- coverage(g, A.or.r = 200, reg.coef = c(10, 2), nsim = 100)

  ##############################################
  # If we know a mean of the prior distribution, #
  ##############################################

  g <- gbp(y, se, mean.PriorDist = 8, model = "gaussian")

  ### when we want to simulate pseudo datasets considering the estimated values
```

```
   ### as true ones.
   gcv <- coverage(g, nsim = 10)

   ### gcv$coverageRB, gcv$coverageS, gcv$average.coverageRB, gcv$average.coverageS,
   ### gcv$minimum.coverageRB, gcv$raw.resultRB, gcv$raw.resultS

   ### gcv <- coverage(g, mean.PriorDist = 3, nsim = 100)
   ### gcv <- coverage(g, A.or.r = 150, nsim = 100)
   ### gcv <- coverage(g, A.or.r = 150, mean.PriorDist = 3, nsim = 100)

################################################################
# Binomial Regression Interactive Multi-level Modeling (BRIMM) #
################################################################

   #################################################################################
   # If we do not have any covariate and do not know a mean of the prior distribution #
   #################################################################################

   b <- gbp(z, n, model = "binomial")

   ### when we want to simulate pseudo datasets considering the estimated values
   ### as true ones.
   bcv <- coverage(b, nsim = 10)

   ### bcv$coverageRB, bcv$coverageS, bcv$average.coverageRB, bcv$average.coverageS,
   ### bcv$minimum.coverageRB, bcv$raw.resultRB, bcv$raw.resultS

   ### bcv <- coverage(b, mean.PriorDist = 0.2, nsim = 100)
   ### bcv <- coverage(b, A.or.r = 50, nsim = 100)
   ### bcv <- coverage(b, reg.coef = -1.5, nsim = 100)
   ### bcv <- coverage(b, A.or.r = 50, mean.PriorDist = 0.2, nsim = 100)
   ### bcv <- coverage(b, A.or.r = 50, reg.coef = -1.5, nsim = 100)

   ##############################################################################
   # If we have one covariate and do not know a mean of the prior distribution yet, #
   ##############################################################################

   b <- gbp(z, n, x1, model = "binomial")

   ### when we want to simulate pseudo datasets considering the estimated values
   ### as true ones.
   bcv <- coverage(b, nsim = 10)

   ### bcv$coverageRB, bcv$coverageS, bcv$average.coverageRB, bcv$average.coverageS,
   ### bcv$minimum.coverageRB, bcv$raw.resultRB, bcv$raw.resultS

   ### bcv <- coverage(b, mean.PriorDist = 0.2, nsim = 100)
   ### bcv <- coverage(b, A.or.r = 50, nsim = 100)
   ### bcv <- coverage(b, reg.coef = c(-1.5, 0), nsim = 100)
   ### bcv <- coverage(b, A.or.r = 40, mean.PriorDist = 0.2, nsim = 100)
   ### bcv <- coverage(b, A.or.r = 40, reg.coef = c(-1.5, 0), nsim = 100)

   ###############################################
```

```
    # If we know a mean of the prior distribution, #
    ################################################

    b <- gbp(z, n, mean.PriorDist = 0.265, model = "binomial")

    ### when we want to simulate pseudo datasets considering the estimated values
    ### as true ones.
    bcv <- coverage(b, nsim = 10)

    ### bcv$coverageRB, bcv$coverageS, bcv$average.coverageRB, bcv$average.coverageS,
    ### bcv$minimum.coverageRB, bcv$raw.resultRB, bcv$raw.resultS

    ### bcv <- coverage(b, mean.PriorDist = 0.2, nsim = 100)
    ### bcv <- coverage(b, A.or.r = 50, nsim = 100)
    ### bcv <- coverage(b, A.or.r = 40, mean.PriorDist = 0.2, nsim = 100)

  ################################################################
  # Poisson Regression Interactive Multi-level Modeling (PRIMM) #
  ################################################################

    ####################################################################################
    # If we do not have any covariate and do not know a mean of the prior distribution #
    ####################################################################################

    p <- gbp(z, n, model = "poisson")

    ### when we want to simulate pseudo datasets considering the estimated values
    ### as true ones.
    pcv <- coverage(p, nsim = 10)

    ### pcv$coverageRB, pcv$coverageS, pcv$average.coverageRB, pcv$average.coverageS,
    ### pcv$minimum.coverageRB, pcv$raw.resultRB, pcv$raw.resultS

    ### pcv <- coverage(p, mean.PriorDist = 0.265, nsim = 100)
    ### pcv <- coverage(p, A.or.r = 120, nsim = 100)
    ### pcv <- coverage(p, reg.coef = -2, nsim = 100)
    ### pcv <- coverage(p, A.or.r = 120, mean.PriorDist = 0.265, nsim = 100)
    ### pcv <- coverage(p, A.or.r = 120, reg.coef = -2, nsim = 100)

    ###############################################################################
    # If we have one covariate and do not know a mean of the prior distribution yet, #
    ###############################################################################

    p <- gbp(z, n, x1, model = "poisson")

    ### when we want to simulate pseudo datasets considering the estimated values
    ### as true ones.
    pcv <- coverage(p, nsim = 10)

    ### pcv$coverageRB, pcv$coverageS, pcv$average.coverageRB, pcv$average.coverageS,
    ### pcv$minimum.coverageRB, pcv$raw.resultRB, pcv$raw.resultS

    ### pcv <- coverage(p, mean.PriorDist = 0.265, nsim = 100)
```

```
### pcv <- coverage(p, A.or.r = 170, nsim = 100)
### pcv <- coverage(p, reg.coef = c(-2, 1), nsim = 100)
### pcv <- coverage(p, A.or.r = 170, mean.PriorDist = 0.265, nsim = 100)
### pcv <- coverage(p, A.or.r = 170, reg.coef = c(-2, 1), nsim = 100)

#################################################
# If we know a mean of the prior distribution, #
#################################################

p <- gbp(z, n, mean.PriorDist = 0.265, model = "poisson")

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
pcv <- coverage(p, nsim = 10)

### pcv$coverageRB, pcv$coverageS, pcv$average.coverageRB, pcv$average.coverageS,
### pcv$minimum.coverageRB, pcv$raw.resultRB, pcv$raw.resultS

### pcv <- coverage(p, mean.PriorDist = 0.265, nsim = 100)
### pcv <- coverage(p, A.or.r = 150, nsim = 100)
### pcv <- coverage(p, A.or.r = 150, mean.PriorDist = 0.265, nsim = 100)
```

---

gbp   *Fitting Gaussian, Poisson, and Binomial Hierarchical Models*

Description

gbp fits Bayesian hierarchical models using the Uniform distribution on the second level variance component (variance of the prior distribution), which enables good frequentist repeated sampling properties.

Usage

```
## Default S3 method:
gbp(x, w, covariates, mean.PriorDist, model = "gaussian", intercept = TRUE,
                        Alpha = 0.95, n.IS = 0, trial.scale = 2.5)
```

Arguments

x	a (k by 1) vector of k groups' sample means for Gaussian or of each group's number of successful trials for Binomial and Poisson data, where k is the number of groups (or units) in a dataset.
w	a (k by 1) vector composed of the standard errors of all groups for Gaussian or of each group's total number of trials for Binomial and Poisson data.
covariates	(optional) a matrix of covariate(s) each column of which corresponds to one covariate.

mean.PriorDist (optional) a numeric value for the second-level mean parameter, *i.e.* the mean of prior distribution, if you know this value a priori.

model   a character string indicating which hierarchical model to fit. "gaussian" for Gaussian data, "poisson" for Poisson, and "binomial" for Binomial. Default is "gaussian"

intercept   TRUE or FALSE flag indicating whether an intercept should be included in the regression. Default is TRUE.

Alpha   a float between 0 and 1 to estimate 100*Alpha% intervals. Default is 0.95.

n.IS   If n.IS = 5000, all the results will be based on 5000 importance samples. Default is 0.

trial.scale   Scale in trial distribution where α is sampled. If resultant weight has too mamy 0's, scale should be smaller than before. If resultant weight has too few 0's, scale should be bigger than before. If there are relatively huge weights, scale should be bigger than before. Default is 2.5.

Details

gbp fits a hierarchical model whose first-level hierarchy is a distribution of observed data and second-level is a conjugate prior distribution on the first-level parameter. To be specific, for Normal data, gbp constructs a two-level Normal-Normal multilevel model. V_j ($=\sigma^2/n_j$) is assumed to be known or to be accurately estimated, and subscript *j* indicates *j*-th group (or unit) in a dataset.

$$(y_j|\theta_j) \sim indep N(\theta_j, V_j)$$

$$(\theta_j|\mu_{0j}, A) \sim indep N(\mu_{0j}, A)$$

$$\mu_{0j} = x'_j\beta$$

for $j = 1, \ldots, k$, where *k* is the number of groups (units) in a dataset.

For Poisson data, gbp builds a two-level Poisson-Gamma multilevel model. A square bracket below indicates [mean, variance] of distribution, a constant multiplied to the notation representing Gamma distribution (Gam) is a scale, and $y_j = \frac{z_j}{n_j}$.

$$(z_j|\theta_j) \sim indep Pois(n_j\theta_j)$$

$$(\theta_j|r, \mu_{0j}) \sim indep \frac{1}{r} Gam(r\mu_{0j}) \sim indep Gam[\mu_{0j}, \mu_{0j}/r]$$

$$log(\mu_{0j}) = x'_j\beta$$

for $j = 1, \ldots, k$, where *k* is the number of groups (units) in a dataset.

For Binomial data, gbp sets a two-level Binomial-Beta multilevel model. A square bracket below indicates [mean, variance] of distribution and $y_j = \frac{z_j}{n_j}$.

$$(z_j|\theta_j) \sim indep Bin(n_j, \theta_j)$$

$$(\theta_j|r, \mu_{0j}) \sim indep Beta(r\mu_{0j}, r(1 - \mu_{0j})) \sim indep Beta[\mu_{0j}, \mu_{0j}(1 - \mu_{0j})/(r + 1)]$$

$$logit(\mu_{0j}) = x'_j\beta$$

for $j = 1, \ldots, k$, where *k* is the number of groups (units) in a dataset.

For reference, based on the above notations, the Uniform prior distribution on the second level variance component (variance of the prior distribution) is *dA* for Gaussian and $d(\frac{1}{r})$ $(= \frac{dr}{r^2})$ for Binomial and Poisson data. The second level variance component can be interpreted as variation among the first-level parameters (θ_j) or variance of ensemble information.

Under this setting, the argument x in gbp is a (*k* by 1) vector of *k* groups' sample means ($y'_j s$ in the description of Normal-Normal model above) for Gaussian or of each group's number of successful trials ($z'_j s$) for Binomial and Poisson data, where *k* is the number of groups (or units) in a dataset.

The argument w is a (*k* by 1) vector composed of the standard errors ($V'_j s$) of all groups for Gaussian or of each group's total number of trials ($n'_j s$) for Binomial and Poisson data.

As for two optional arguments, `covariates` and `mean.PriorDist`, there are three feasible combinations of them to run gbp. The first situation is when we do not have any covariate and do not know a mean of the prior distribution (μ_0) a priori. In this case, assigning none of two optional arguments, such as `"gbp(z, n, model = "binomial")"`, will lead to a correct model. gbp will automatically fit a regression with only an intercept term to estimate a common mean of the prior distribution (exchangeability).

The second situation is when we have covariate(s) and do not know a mean of the prior distribution (μ_0) a priori. In this case, assigning a matrix, *X*, each column of which corresponds to one covariate, such as `"gbp(z, n, X, model = "poisson")"`, will lead to a correct model. Default of gbp is to fit a regression including an intercept term to estimate a mean of the prior distribution. Double exchangeability will hold in this case.

The last case is when we know a mean of the prior distribution (μ_0) a priori. Now, we do not need to estimate regression coefficients at all because we know a true value of μ_0 (strong assumption). Designating this value into the argument of gbp like `"gbp(y, se, mean.PriorDist = 3)"` is enough to account for it. For reference, `mean.PriorDist` has a stronger priority than `covariates`, which means that when both arguments are designated, gbp will fit a hierarchical model using the known mean of prior distribution, `mean.PriorDist`.

gbp returns an object of class "gbp" which provides three relevant functions `plot`, `print`, and `summary`.

Value

An object of class gbp comprises of:

`sample.mean`	sample mean of each group (or unit)
`se`	if Gaussian data, standard error of sample mean in each group (or unit)
`n`	if Binomial and Poisson data, total number of trials of each group (or unit)
`prior.mean`	numeric if entered, NA if not entered
`prior.mean.hat`	estimate of prior mean by a regression if prior mean is not assigned a priori
`shrinkage`	shrinkage estimate of each group (adjusted posterior mean)
`sd.shrinkage`	posterior standard deviation of shrinkage
`post.mean`	posterior mean of each group
`post.sd`	posterior standard deviation of each group
`post.intv.low`	lower bound of 100*Alpha% posterior interval (quantile of posterior distribution)

post.intv.upp	upper bound of 100*Alpha% posterior interval (quantile of posterior distribution)
model	"gaussian" for Gaussian, "poisson" for Poisson, and "binomial" for Binomial data
X	a covariate vector or matrix if designated. NA if not
beta.new	regression coefficient estimates
beta.var	estimated variance matrix of regression coefficient
intercept	whether TRUE or FALSE
a.new	a posterior mode of α defined as $\log(A)$ for Gaussian or $\log(\frac{1}{r})$ for Binomial and Poisson data. Practical meaning (variation of ensemble information) of estimating α will appear in summary(gbp.object).
a.var	posterior variance of α
Alpha	confidence level based on which confidence interval is constructed
weight	importance weights, if importance sampling is done

Author(s)

Joseph Kelly, Hyungsuk Tak, and Carl Morris

References

Morris, C. and Lysy, M. (2012). Shrinkage Estimation in Multilevel Normal Models. *Statistical Science*. **27**. 115-134.

Examples

```
# Loading datasets
data(schools)
y <- schools$y
se <- schools$se

# Arbitrary covariate for schools data
x2 <- rep(c(-1, 0, 1, 2), 2)

# baseball data where z is Hits and n is AtBats
z <- c(18, 17, 16, 15, 14, 14, 13, 12, 11, 11, 10, 10, 10, 10, 10,  9,  8,  7)
n <- c(45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45)

# One covariate: 1 if a player is an outfielder and 0 otherwise
x1 <- c(1,  1,  1,  1,  1,  0,  0,  0,  0,  1,  0,  0,  0,  1,  1,  0,  0,  0)

###############################################################
# Gaussian Regression Interactive Multilevel Modeling (GRIMM) #
###############################################################

  ############################################################################
  # If we do not have any covariate and do not know a mean of the prior distribution #
```

```
################################################################################

g <- gbp(y, se, model = "gaussian")
g
print(g, sort = FALSE)
summary(g)
plot(g)
plot(g, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
gcv <- coverage(g, nsim = 10)

### gcv$coverageRB, gcv$coverage10, gcv$average.coverageRB, gcv$average.coverage10,
### gcv$minimum.coverageRB, gcv$minimum.coverage10, gcv$raw.resultRB, gcv$raw.result10.

### when we want to simulate pseudo datasets based on different values of A
### and a regression coefficient (intercept),
### not using estimated values as true ones.
gcv <- coverage(g, A.or.r = 9, reg.coef = 10, nsim = 10)

################################################################################
# If we have one covariate and do not know a mean of the prior distribution yet, #
################################################################################

g <- gbp(y, se, x2, model = "gaussian")
g
print(g, sort = FALSE)
summary(g)
plot(g)
plot(g, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
gcv <- coverage(g, nsim = 10)

### gcv$coverageRB, gcv$coverage10, gcv$average.coverageRB, gcv$average.coverage10,
### gcv$minimum.coverageRB, gcv$minimum.coverage10, gcv$raw.resultRB, gcv$raw.result10.

### when we want to simulate pseudo datasets based on different values of A
### and regression coefficients, not using estimated values
### as true ones. Two values of reg.coef are for beta0 and beta1
gcv <- coverage(g, A.or.r = 9, reg.coef = c(10, 1), nsim = 10)

###############################################
# If we know a mean of the prior distribution, #
###############################################

g <- gbp(y, se, mean.PriorDist = 8, model = "gaussian")
g
print(g, sort = FALSE)
summary(g)
plot(g)
```

```
    plot(g, sort = FALSE)

    ### when we want to simulate pseudo datasets considering the estimated values
    ### as true ones.
    gcv <- coverage(g, nsim = 10)

    ### gcv$coverageRB, gcv$coverage10, gcv$average.coverageRB, gcv$average.coverage10,
    ### gcv$minimum.coverageRB, gcv$minimum.coverage10, gcv$raw.resultRB, gcv$raw.result10.

    ### when we want to simulate pseudo datasets based on different values of A and
    ### 2nd level mean as true ones, not using estimated values as true ones.
    coverage(g, A.or.r = 9, mean.PriorDist = 5, nsim = 10)

##############################################################
# Binomial Regression Interactive Multilevel Modeling (BRIMM) #
##############################################################

    #####################################################################################
    # If we do not have any covariate and do not know a mean of the prior distribution #
    #####################################################################################

    b <- gbp(z, n, model = "binomial")
    b
    print(b, sort = FALSE)
    summary(b)
    plot(b)
    plot(b, sort = FALSE)

    ### when we want to simulate pseudo datasets considering the estimated values
    ### as true ones.
    bcv <- coverage(b, nsim = 10)

    ### bcv$coverageRB, bcv$coverage10, bcv$average.coverageRB, bcv$average.coverage10,
    ### bcv$minimum.coverageRB, bcv$minimum.coverage10, bcv$raw.resultRB, bcv$raw.result10.

    ### when we want to simulate pseudo datasets based on different values of r
    ### and a regression coefficient (intercept),
    ### not using estimated values as true ones.
    bcv <- coverage(b, A.or.r = 60, reg.coef = -1, nsim = 10)

    #####################################################################################
    # If we have one covariate and do not know a mean of the prior distribution yet, #
    #####################################################################################

    b <- gbp(z, n, x1, model = "binomial")
    b
    print(b, sort = FALSE)
    summary(b)
    plot(b)
    plot(b, sort = FALSE)

    ### when we want to simulate pseudo datasets considering the estimated values
    ### as true ones.
```

```
bcv <- coverage(b, nsim = 10)

### bcv$coverageRB, bcv$coverage10, bcv$average.coverageRB, bcv$average.coverage10,
### bcv$minimum.coverageRB, bcv$minimum.coverage10, bcv$raw.resultRB, bcv$raw.result10.

### when we want to simulate pseudo datasets based on different values of r
### and regression coefficients, not using estimated values
### as true ones. Two values of reg.coef are for beta0 and beta1
bcv <- coverage(b, A.or.r = 60, reg.coef = c(-1, 0), nsim = 10)

#############################################
# If we know a mean of the prior distribution, #
#############################################

b <- gbp(z, n, mean.PriorDist = 0.265, model = "binomial")
b
print(b, sort = FALSE)
summary(b)
plot(b)
plot(b, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
bcv <- coverage(b, nsim = 10)

### bcv$coverageRB, bcv$coverage10, bcv$average.coverageRB, bcv$average.coverage10,
### bcv$minimum.coverageRB, bcv$minimum.coverage10, bcv$raw.resultRB, bcv$raw.result10.

### when we want to simulate pseudo datasets based on different values of r and
### 2nd level mean as true ones, not using estimated values as true ones.
bcv <- coverage(b, A.or.r = 60, mean.PriorDist = 0.3, nsim = 10)

##############################################################
# Poisson Regression Interactive Multilevel Modeling (PRIMM) #
##############################################################

##################################################################################
# If we do not have any covariate and do not know a mean of the prior distribution #
##################################################################################

p <- gbp(z, n, model = "poisson")
p
print(p, sort = FALSE)
summary(p)
plot(p)
plot(p, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
pcv <- coverage(p, nsim = 10)

### pcv$coverageRB, pcv$coverage10, pcv$average.coverageRB, pcv$average.coverage10,
### pcv$minimum.coverageRB, pcv$minimum.coverage10, pcv$raw.resultRB, pcv$raw.result10.
```

```
### when we want to simulate pseudo datasets based on different values of r
### and a regression coefficient (intercept),
### not using estimated values as true ones.
pcv <- coverage(p, A.or.r = 60, reg.coef = -5, nsim = 10)

################################################################################
# If we have one covariate and do not know a mean of the prior distribution yet, #
################################################################################

p <- gbp(z, n, x1, model = "poisson")
p
print(p, sort = FALSE)
summary(p)
plot(p)
plot(p, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
pcv <- coverage(p, nsim = 10)

### pcv$coverageRB, pcv$coverage10, pcv$average.coverageRB, pcv$average.coverage10,
### pcv$minimum.coverageRB, pcv$minimum.coverage10, pcv$raw.resultRB, pcv$raw.result10.

### when we want to simulate pseudo datasets based on different values of r
### and regression coefficients, not using estimated values
### as true ones. Two values of reg.coef are for beta0 and beta1
pcv <- coverage(p, A.or.r = 60, reg.coef = c(-2, 0), nsim = 10)

################################################
# If we know a mean of the prior distribution, #
################################################

p <- gbp(z, n, mean.PriorDist = 0.265, model = "poisson")
p
print(p, sort = FALSE)
summary(p)
plot(p)
plot(p, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
pcv <- coverage(p, nsim = 10)

### pcv$coverageRB, pcv$coverage10, pcv$average.coverageRB, pcv$average.coverage10,
### pcv$minimum.coverageRB, pcv$minimum.coverage10, pcv$raw.resultRB, pcv$raw.result10.

### when we want to simulate pseudo datasets based on different values of r and
### 2nd level mean as true ones, not using estimated values as true ones.
pcv <- coverage(p, A.or.r = 60, mean.PriorDist = 0.3, nsim = 10)
```

---

hospital   *Thirty-one Hospital Data*

Description

Medical profiling evaluation of 31 New York hospitals in 1992. We are to consider these as Normally-distributed indices of successful outcome rates for patients at these 31 hospitals following Coronary Artery Bypass Graft (CABG) surgeries. The indices are centered so that the New York statewide average outcome over all hospitals lies near 0. Larger estimates of y indicate hospitals that performed better for these surgeries.

Usage

```
data(hospital)
```

Format

A dataset of 31 hospitals comprises of:

y values obtained through a variance stabilizing transformation of the unbiased death rate estimates, d / n, assuming Binomial data. Details in the reference.

se approximated standard error of y.

d the number of deaths within a month of CABG surgeries in each hospital

n total number of patients receiving CABG surgeries (case load) in each hospital

Source

Morris, C. and Lysy, M. (2012). Shrinkage Estimation in Multilevel Normal Models. *Statistical Science*. **27**. 115-134.

Examples

```
data(hospital)
```

---

plot.gbp   *Drawing Shrinkage and Posterior Interval Plots*

Description

`plot(gbp.object)` draws shrinkage and posterior interval plots

Usage

```
## S3 method for class 'gbp'
plot(x, sort = TRUE, ...)
```

Arguments

    x   a resultant object of gbp function.

    sort   TRUE or FALSE flag. If TRUE, the interval plot (second plot) will be drawn by the order of se for Gaussian, or of n for Binomial and Poisson data. If FALSE, it will be by the order of data input. Default is TRUE.

    ...   further arguments passed to other methods.

Details

As for the argument x, if the result of gbp is designated to b like "b <- gbp(z, n, model = "binomial")", the argument x is supposed to be b.

This function produces two plots containing information about the prior, sample, and posterior means.

The first plot is a shrinkage plot representing sample means (black circle) on the upper horizontal line and prior (blue line) and posterior means (red circle) on the lower horizontal line. The aim of this plot is to get a sense of the magnitude of the shrinkage and to observe if any change in ordering of the groups has occurred. Crossovers (changes of order) are noted by a black square as indicated in the legend. If the points plotted have the same value then a sunflower plot is produced where each petal (line protruding from the point) represent the count of points with that value. The plot also aims to incorporate the uncertainty and the lengths of the violet and green lines are proportional to the standard error and the posterior standard deviation respectively.

The final plot shows interval estimates of all the groups (units) in a dataset. Two short horizontal ticks at both ends of each black vertical line indicate 97.5% and 2.5% quantiles of a posterior distribution for each group (Normal for Gaussian, Beta for Binomial, and Gamma for Poisson). Red dots (posterior mean) are between black circles (sample mean) and blue line(s) (prior mean) as a result of shrinkage (regression toward the mean).

If we want to see the interval plot (the second plot) NOT sorted by the order of se for Gaussian, or of n for Binomial and Poisson data, plot(b, sort = FALSE) will show this plot by the order of data input.

Value

Two plots described in *details* will be displayed.

Author(s)

Joseph Kelly, Hyungsuk Tak, and Carl Morris

Examples

```
data(hospital)

z <- hospital$d
n <- hospital$n
y <- hospital$y
se <- hospital$se
```

```
##############################################################################
# We do not have any covariates and do not know a mean of the prior distribution. #
##############################################################################

  ##############################################################
  # Gaussian Regression Interactive Multilevel Modeling (GRIMM) #
  ##############################################################

  g <- gbp(y, se, model = "gaussian")
  plot(g)
  plot(g, sort = FALSE)

  ##############################################################
  # Binomial Regression Interactive Multilevel Modeling (BRIMM) #
  ##############################################################

  b <- gbp(z, n, model = "binomial")
  plot(b)
  plot(b, sort = FALSE)

  ##############################################################
  # Poisson Regression Interactive Multilevel Modeling (PRIMM) #
  ##############################################################

  p <- gbp(z, n, model = "poisson")
  plot(p)
  plot(p, sort = FALSE)
```

---

print.gbp	*Displaying 'gbp' Class*

Description

`print.gbp` enables users to see a compact group-level (unit-level) estimation result of gbp function.

Usage

```
## S3 method for class 'gbp'
print(x, sort = TRUE, ...)
```

Arguments

x	a resultant object of gbp function.
sort	TRUE or FALSE flag. If TRUE, the result will appear by the order of se for Gaussian, or of n for Binomial and Poisson data. If FALSE, it will do by the order of data input. Default is TRUE.
...	further arguments passed to other methods.

Details

As for the argument x, if the result of gbp is designated to b like "b <- gbp(z, n, model = "binomial")",
the argument x is supposed to be b.

We do not need to type "print(b, sort = TRUE)" but "b" itself is enough to call print(b, sort = TRUE).
But if we want to see the result NOT sorted by the order of se for Gaussian, or of n for Binomial
and Poisson data, print(b, sort = FALSE) will show the result by the order of data input.

Value

print(gbp.object) will display:

obs.mean	sample mean of each group
se	if Gaussian data, standard error of each group
n	if Binomial or Poisson data, total number of trials of each group
X	a covariate vector or matrix if designated. NA if not
prior.mean	numeric if entered, NA if not entered
prior.mean.hat	estimate of prior mean by a regression if prior mean is not assigned a priori. The variable name on the display will be "prior.mean"
shrinkage	shrinkage estimate of each group (adjusted posterior mean)
low.intv	lower bound of 100*Alpha% posterior interval
post.mean	posterior mean of each group
upp.intv	upper bound of 100*Alpha% posterior interval
post.sd	posterior standard deviation of each group

Author(s)

Joseph Kelly, Hyungsuk Tak, and Carl Morris

Examples

```
data(hospital)

z <- hospital$d
n <- hospital$n
y <- hospital$y
se <- hospital$se

################################################################################
# We do not have any covariates and do not know a mean of the prior distribution. #
################################################################################

  ############################################################
  # Gaussian Regression Interactive Multilevel Modeling (GRIMM) #
  ############################################################

  g <- gbp(y, se, model = "gaussian")
```

```
g
print(g, sort = FALSE)

##################################################################
# Binomial Regression Interactive Multilevel Modeling (BRIMM) #
##################################################################

b <- gbp(z, n, model = "binomial")
b
print(b, sort = FALSE)

################################################################
# Poisson Regression Interactive Multilevel Modeling (PRIMM) #
################################################################

p <- gbp(z, n, model = "poisson")
p
print(p, sort = FALSE)
```

---

print.summary.gbp   *Displaying 'summary.gbp' Class*

Description

summary(gbp.object) enables users to see a compact summary of estimation result.

Usage

```
## S3 method for class 'summary.gbp'
print(x, ...)
```

Arguments

x	a resultant object of gbp function.
...	further arguments passed to other methods.

Details

The summary has three parts depending on the model fitted by gbp; Main Summary, Second-level Variance Component Estimation Summary, and Regression Summary (if fitted).

A display of Main Summary changes depending on whether all the groups (units) has the same standard error for Gaussian data (or the same total number of trials for Binomial and Poisson data). If they are not the same, Main Summary lists groups (units) with minimum, median, and maximum values of the standard error for Gaussian data (or of the total number of trials for Binomial and Poisson data). And the last row of Main Summary is about the overall average for all the groups (units) within each column. Note that this last row is not an average over displayed groups (units) above.

If groups (units) have the same standard error for Gaussian (or the same total number of trials for Binomial and Poisson), `Main Summary` lists groups (units) with minimum, median, and maximum values of the sample mean.

For reference, if there are several units with the same median value, they will show up with numbering.

The second part is about the `Second-level Variance Component Estimation Summary`. For reference, the second level variance component can be interpreted as variation among the first-level parameters (θ_j) or variance in ensemble information. It is A for Gaussian, $\frac{\mu_{0j}}{r}$ for Poisson, and $\frac{\mu_{0j}(1-\mu_{0j})}{r}$ for Binomial data. To be specific, this part shows estimate of α (a posterior mode) defined as $\log(A)$ for Gaussian or $\log(\frac{1}{r})$ for Binomial and Poisson data, and its standard error.

The last part depends on whether gbp fitted a regression or not. For reference, gbp fits a regression if the second-level mean (`mean.PriorDist`) was not designated. In this case, `summary(gbp.object)` will display the result of regression fit.

Value

`summary(gbp.object)` shows a compact summary of estimation result such as:

Main summary   **Unit w/ min(se or n)** an estimation result of a group (unit) with the minimum standard error for Gaussian or the minimum total number of trials for Binomial and Poisson data.

Unit w/ min(sample.mean) appears instead of `Group w/ min(se or n)` when all the groups (units) have the same standard error for Gaussian or the same total number of trials for Binomial and Poisson data.

Unit w/ median(se or n) an estimation result of group(s) (unit(s)) with the median standard error for Gaussian or the median total number of trials for Binomial and Poisson data.

Unit w/ median(sample.mean) appears instead of `Group w/ median(se or n)` when all the groups (units) have the same standard error for Gaussian or the same total number of trials for Binomial and Poisson data.

Unit w/ max(se or n) an estimation result of a group (unit) with the maximum standard error for Gaussian or the maximum total number of trials for Binomial and Poisson data.

Unit w/ max(sample.mean) appears instead of `Group w/ max(se or n)` when all the groups (units) have the same standard error for Gaussian or the same total number of trials for Binomial and Poisson data.

Overall Means the overall average for all the groups (units) within each column.

Second-level Variance Component Estimation Summary

post.mode.alpha a posterior mode of α defined as $\log(A)$ for Gaussian or $\log(\frac{1}{r})$ for Binomial and Poisson data.

post.sd.alpha standard deviation of the posterior distribution of alpha

Regression Summary (if fitted)

estimate regression coefficient estimates.

se estimated standard error of regression coefficients.

z.val estimate / se.

p.val two-sided p-values.

Author(s)

Joseph Kelly, Hyungsuk Tak, and Carl Morris

Examples

```
data(hospital)

z <- hospital$d
n <- hospital$n
y <- hospital$y
se <- hospital$se

################################################################################
# We do not have any covariates and do not know a mean of the prior distribution. #
################################################################################

  #############################################################
  # Gaussian Regression Interactive Multilevel Modeling (GRIMM) #
  #############################################################

  g <- gbp(y, se, model = "gaussian")
  summary(g)

  #############################################################
  # Binomial Regression Interactive Multilevel Modeling (BRIMM) #
  #############################################################

  b <- gbp(z, n, model = "binomial")
  summary(b)

  #############################################################
  # Poisson Regression Interactive Multilevel Modeling (PRIMM) #
  #############################################################

  p <- gbp(z, n, model = "poisson")
  summary(p)
```

Rgbp	*Gaussian, Poisson, and Binomial Hierarchical Modeling*

Description

Bayesian-frequentist reconciliation via approximate Bayesian hierarchical modeling and frequency method checking for Gaussian, Binomial, and Poisson data.

Details

Package:	Rgbp
Type:	Package
Version:	1.0.0
Date:	2013-03-16
License:	GPL-2
Main functions:	gbp, coverage

Rgbp is an R package that utilizes approximate Bayesian machinery to provide a method of estimating two-level hierarchical models for Gaussian, Poisson, and Binomial data in a fast and computationally efficient manner. The main products of this package are point and interval estimates for the true parameters, whose good frequency properties can be validated via its repeated sampling procedure called frequency method checking. It is found that such Bayesian-frequentist reconciliation allows Rgbp to have attributes desirable from both perspectives, working well in small samples and yielding good coverage probabilities for its interval estimates.

Author(s)

Joseph Kelly, Hyungsuk Tak, and Carl Morris

Maintainer: Joseph Kelly <kelly2@fas.harvard.edu>

References

Morris, C. and Lysy, M. (2012). Shrinkage Estimation in Multilevel Normal Models. *Statistical Science*. **27**. 115-134.

Examples

```
# Loading datasets
data(schools)
y <- schools$y
se <- schools$se

# Arbitrary covariate for schools data
x2 <- rep(c(-1, 0, 1, 2), 2)

# baseball data where z is Hits and n is AtBats
z <- c(18, 17, 16, 15, 14, 14, 13, 12, 11, 11, 10, 10, 10, 10, 10,  9,  8,  7)
n <- c(45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45)

# One covariate: 1 if a player is an outfielder and 0 otherwise
x1 <- c(1,  1,  1,  1,  1,  0,  0,  0,  0,  1,  0,  0,  0,  1,  1,  0,  0,  0)

################################################################
# Gaussian Regression Interactive Multilevel Modeling (GRIMM) #
################################################################

  ##############################################################################
  # If we do not have any covariate and do not know a mean of the prior distribution #
```

```
#####################################################################

g <- gbp(y, se, model = "gaussian")
g
print(g, sort = FALSE)
summary(g)
plot(g)
plot(g, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
### gcv <- coverage(g, nsim = 10)
### more details in ?coverage

#####################################################################
# If we have one covariate and do not know a mean of the prior distribution yet, #
#####################################################################

g <- gbp(y, se, x2, model = "gaussian")
g
print(g, sort = FALSE)
summary(g)
plot(g)
plot(g, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
### gcv <- coverage(g, nsim = 10)
### more details in ?coverage

###############################################
# If we know a mean of the prior distribution, #
###############################################

g <- gbp(y, se, mean.PriorDist = 8, model = "gaussian")
g
print(g, sort = FALSE)
summary(g)
plot(g)
plot(g, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
### gcv <- coverage(g, nsim = 10)
### more details in ?coverage

###############################################################
# Binomial Regression Interactive Multilevel Modeling (BRIMM) #
###############################################################

#####################################################################
# If we do not have any covariate and do not know a mean of the prior distribution #
#####################################################################
```

```
b <- gbp(z, n, model = "binomial")
b
print(b, sort = FALSE)
summary(b)
plot(b)
plot(b, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
### bcv <- coverage(b, nsim = 10)
### more details in ?coverage


################################################################################
# If we have one covariate and do not know a mean of the prior distribution yet, #
################################################################################

b <- gbp(z, n, x1, model = "binomial")
b
print(b, sort = FALSE)
summary(b)
plot(b)
plot(b, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
### bcv <- coverage(b, nsim = 10)
### more details in ?coverage


################################################
# If we know a mean of the prior distribution, #
################################################

b <- gbp(z, n, mean.PriorDist = 0.265, model = "binomial")
b
print(b, sort = FALSE)
summary(b)
plot(b)
plot(b, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
### bcv <- coverage(b, nsim = 10)
### more details in ?coverage

###############################################################
# Poisson Regression Interactive Multilevel Modeling (PRIMM) #
###############################################################

################################################################################
# If we do not have any covariate and do not know a mean of the prior distribution #
################################################################################
```

```
p <- gbp(z, n, model = "poisson")
p
print(p, sort = FALSE)
summary(p)
plot(p)
plot(p, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
### pcv <- coverage(p, nsim = 10)
### more details in ?coverage

##############################################################################
# If we have one covariate and do not know a mean of the prior distribution yet, #
##############################################################################

p <- gbp(z, n, x1, model = "poisson")
p
print(p, sort = FALSE)
summary(p)
plot(p)
plot(p, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
### pcv <- coverage(p, nsim = 10)
### more details in ?coverage

################################################
# If we know a mean of the prior distribution, #
################################################

p <- gbp(z, n, mean.PriorDist = 0.265, model = "poisson")
p
print(p, sort = FALSE)
summary(p)
plot(p)
plot(p, sort = FALSE)

### when we want to simulate pseudo datasets considering the estimated values
### as true ones.
### pcv <- coverage(p, nsim = 10)
### more details in ?coverage
```

---

schools   *Eight Schools Data*

Description

Dataset as seen in Rubin (1981) which was an analysis of coaching effects on SAT scores from eight schools.

Usage

```
data(schools)
```

Format

A dataset of 8 schools containing

y The observed coaching effect of each school

se The standard error of the coaching effect of each school.

Source

Rubin, D. B. (1981). *Estimation in parallel randomized experiments.* Journal of Educational Statistics, 6:377-401.

Examples

```
data(schools)
```

---

summary.gbp   *Summarizing Estimation Result*

Description

summary.gbp prepares a summary of estimation result saved in the object defined as "gbp" class creating "summary.gbp" class

Usage

```
## S3 method for class 'gbp'
summary(object, ...)
```

Arguments

object   a resultant object of gbp function.

...   further arguments passed to other methods.

Value

summary.gbp prepares below contents:

main	a table to be displayed by summary(gbp.object). print.summary.gbp.
sec.var	a vector containing an estimation result of the second-level variance component. print.summary.gbp.
reg	a vector composed of a summary of regression fit (if fitted). print.summary.gbp.

Author(s)

Joseph Kelly, Carl Morris, and Hyungsuk Tak

Examples

```
data(hospital)

z <- hospital$d
n <- hospital$n
y <- hospital$y
se <- hospital$se

################################################################################
# We do not have any covariates and do not know a mean of the prior distribution. #
################################################################################

  ##############################################################
  # Gaussian Regression Interactive Multilevel Modeling (GRIMM) #
  ##############################################################

  g <- gbp(y, se, model = "gaussian")
  summary(g)

  ##############################################################
  # Binomial Regression Interactive Multilevel Modeling (BRIMM) #
  ##############################################################

  b <- gbp(z, n, model = "binomial")
  summary(b)

  ##############################################################
  # Poisson Regression Interactive Multilevel Modeling (PRIMM) #
  ##############################################################

  p <- gbp(z, n, model = "poisson")
  summary(p)
```

Index