# Package ‘Rgbp’

March 19, 2013

**Version** 1.0.0

**Date** 2013-03-16

**Title** Bayesian Hierarchical Modeling using Generalized Stein’s Harmonic Prior

**Author** Joseph Kelly, Carl Morris, and Hyungsuk Tak

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

**Depends** sn, mnormt

**Description** Bayesian Hierarchical modeling for Gaussian (GRIMM), Binomial (BRIMM) and Poisson (PRIMM) data using generalized Stein’s harmonic prior.

**License** GPL-2

**BugReports** https://github.com/jyklly/gbp/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 10 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

### 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)
z <- baseball$Hits
n <- baseball$At.Bats

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

#########
# BRIMM #
#########

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

```
plot(b)

#########
# PRIMM #
#########

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

---

coverage *Estimating Coverage Probability*

---

## Description

coverage estimates Rao-Blackwellized and unbiased coverage probabilities.

## Usage

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

## Arguments

| | |
|---|---|
| gbp.object | a resultant object of gbp function. |
| A.or.r | (optional) a numeric value of *A* for GRIMM or of *r* for BRIMM (and PRIMM). Designating this argument should come with other arguments, for example, (A.or.r, reg.coef, covariates (if any)) or (A.or.r, mean.PriorDist). |
| reg.coef | (optional) a (*m* by 1) vector for regression coefficients, $\beta$, where *m* is the number of regression coefficients including an intercept. |
| covariates | (optional) a (*k* by *t*) matrix of covariates without a column of ones for an intercept, where *k* is the number of groups (or units) in a dataset and *t* is the number of covariates (t $\geq$ 1). If gbp fits an intercept in the regression, *t = (m - 1)*. |
| mean.PriorDist | (optional) a numeric value for the mean of (second-level) prior distribution. |
| nsim | number of simulations (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 = "br")", the argument gbp.object indicates this b.

Data generating process is based on a second-level hierarchical model. The first-level is a distribution of observed data (Likelihood) and the second-level is a conjugate prior distribution on the first-level parameter. Covariates appear in the second-level because covariates are obtainable before we observe data.

To be specific, for Normal data, gbp constructs a two-level Normal-Normal multilevel model ($\sigma_j^2$ is assumed to be known and subscript *j* indicates *j*-th group (or unit) in a dataset):

$$(y_j \mid \theta_j) \sim indep\, N(\theta_j, \sigma_j^2)$$

$$(\theta_j \mid A,\ \mu_{0j}) \ \sim\ indep\, N(\mu_{0j},\ A)$$
$$\mu_{0j} \ =\ x_j'\beta$$

for $j = 1, \ldots, k$.

For Poisson data, gbp builds a two-level Poisson-Gamma multilevel model (a square bracket below indicates [mean, variance] of distribution):

$$(z_j \mid \theta_j) \ \sim\ indep\, Pois(n_j\theta_j)$$
$$(\theta_j \mid r,\ \mu_{0j}) \ \sim\ indep\, Gam(r\mu_{0j}, scale = 1/r) \ \sim\ indep\, Gam[\mu_{0j}, \mu_{0j}/r]$$
$$log(\mu_{0j}) \ =\ x_j'\beta$$

for $j = 1, \ldots, k$.

For Binomial data, gbp sets a two-level Binomial-Beta multilevel model:

$$(z_j \mid \theta_j) \ \sim\ indep\, Bin(n_j, \theta_j)$$
$$(\theta_j \mid r, \mu_{0j}) \ \sim\ indep\, Beta(r\mu_{0j},\ r(1-\mu_{0j})) \ \sim\ indep\, Beta[\mu_{0j},\ \mu_{0j}(1-\mu_{0j})\ /\ (r+1)]$$
$$logit(\mu_{0j}) \ =\ x_j'\beta$$

for $j = 1, \ldots, k$.

From now on, the subscript *i* means *i*-th simulation and *j* indicates *j*-th group (or unit).

Pseudo-data generating process starts from the second-level to the first-level hiearchy. `coverage` first generates true parameters ($\theta_i$) for *k* groups (units) at the second-level and then moves onto the first-level to simulate pseudo-data sets, $y_i$ for GRIMM or $z_i$ for BRIMM and PRIMM, given previously generated true parametes ($\theta_i$).

So, in order to generate pseudo-datasets, `coverage` needs parameters of prior distribution, *A* (or *r*), $\beta$ (`reg.coef`), and *X* (`covariates`) (if any), or *A* (or *r*) and $\mu_0$ (`mean.PriorDist`) to start the pseudo-data generating process described above. From here, we have four options to run gbp.

First, if any values related to the prior distribution are not designated like `coverage(b, nsim = 10)`, then `coverage` will regard estimated values in `b` (`gbp.object`) as given true values when it generates bunch of pseudo datasets. After sampling a (*k* by 1) vector $\theta_i$ from the prior distribution determined by those estimated values in `b` (`gbp.object`), `coverage` creates an *i*-th pseudo-dataset based on $\theta_i$ just sampled.

Second, `coverage` allows us to designate different true values in generating datasets, for example `coverage(b, A.or.r = 15, reg.coef = 3, nsim = 100)` assuming we do not have any covariates and do not know a mean of the prior distribution. One value designated in `reg.coef` will be used to calculate the mean of second-level distribution by $g(\mu_0) = \beta_0 = 3$. Then, `coverage` samples a (*k* by 1) vector $\theta_i$ from the prior distribution determined by designated values, `A.or.r` and `reg.coef` (only intercept term). Sampling *i*-th pseudo-data is based on $\theta_i$ just sampled.

Third, `coverage` enables us to designate different true values in generating datasets like `coverage(b, A.or.r = 15, reg.coef = c(3, -1), covariates = X, nsim = 100)` when we have one covariate (can be more than one but `reg.coef` should reflect on the number of regression coefficients including an intercept term) without any knowledge about the mean of prior distribution a priori. For reference, a covariate matrix, `X`, should not include a column of ones for an intercept term in the regression and the mean of prior distribution will be set as $g(\mu_{0j}) = x_j'\beta$, where $x_j$ is (1, *j*-th row of `X`)'. Then, `coverage` samples a (*k* by 1) vector $\theta_i$ from the prior distribution determined by designated values, `A.or.r`, `reg.coef`, and `covariates`. Sampling *i*-th pseudo-data is based on $\theta_i$ just sampled.

Lastly, `coverage` provides us a way to designate different true values in generating datasets like `coverage(b, A.or.r = 15, mean.PriorDist = 0.45, nsim = 100)` when we know the mean of prior distribution a priori. Then, `coverage` samples a ($k$ by 1) vector $\theta_i$ from the prior distribution determined by designated values, `A.or.r` and `mean.PriorDist`. The $i$-th Pseudo-datasets are generated based on $\theta_i$ just sampled.

The unbiased estimator of coverage probability in $j$-th group (or unit) 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 $\theta_{ij}$ that generated the observed value of the $j$-th group in the $i$-th dataset.

Rao-Blackwellized estimator is an expectation of the unbiased estimator described above given a sufficient statistic, $y$.

## Value

| | |
|---|---|
| `coverageRB` | estimated Rao-Blackwellized coverage probability for each group (or unit) averaged over all simulations. |
| `coverage10` | estimated unbiased coverage probability for each group (or unit) averaged over all simulations. |
| `average.coverageRB` | average value over `coverageRB`. |
| `average.coverage10` | average value over `coverage10`. |
| `minimum.coverageRB` | minimum value among `coverageRB`. |
| `minimum.coverage10` | minimum value among `coverage10`. |
| `raw.resultRB` | all the Rao-Blackwellized coverage probabilities for every group and for every simulation. |
| `raw.result10` | all the unbiased coverage probabilities for every group and for every simulation. |

## Author(s)

Joseph Kelly, Carl Morris, and Hyungsuk Tak

## 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)

# 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)
x1 <- rep(c(-1, 0, 1), 6)

y <- schools$y
se <- schools$se
x2 <- rep(c(-1, 0, 1, 2), 2)
```

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

#########
# GRIMM #
#########

g <- gbp(y, se)

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(g, nsim = 10)

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

#########
# BRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(b, nsim = 10)

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

#########
# PRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(p, nsim = 10)

### when we want to simulate psuedo datasets based on different values of r and
### of a regression coefficient (intercept),
### not using estimated values as true ones.
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, #
################################################################################

#########
# GRIMM #
#########
```

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(g, nsim = 10)

### when we want to simulate psuedo datasets based on different values of A,
### of regression coefficients, and of covariate, not using estimated values
### as true ones.
coverage(g, A.or.r = 9, reg.coef = c(10, 1), covariates = x2, nsim = 10)
### two values of reg.coef are for beta0 and beta1

#########
# BRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(b, nsim = 10)

### when we want to simulate psuedo datasets based on different values of r,
### of regression coefficients, and of covariate, not using estimated values
### as true ones.
coverage(b, A.or.r = 60, reg.coef = c(-1, 0), covariates = x1, nsim = 10)
### two values of reg.coef are for beta0 and beta1

#########
# PRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(p, nsim = 10)

### when we want to simulate psuedo datasets based on different values of r,
### of regression coefficients, and of covariate, not using estimated values
### as true ones.
coverage(p, A.or.r = 60, reg.coef = c(-2, 0), covariates = x1, nsim = 10)
### two values of reg.coef are for beta0 and beta1

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

#########
# GRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(g, nsim = 10)
```

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

#########
# BRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(b, nsim = 10)

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

#########
# PRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(p, nsim = 10)

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

---

gbp *Fitting Bayesian Hierarchical Models*

---

### Description

gbp is used to fit Bayesian hierarchical models for Gaussian (GRIMM), Binomial (BRIMM), and Poisson (PRIMM) data using generalized Stein's harmonic prior for good frequentist repeated sampling property.

### Usage

```
## Default S3 method:
gbp(x, y, covariates, mean.PriorDist, model = "gr", intercept = TRUE, Alpha = 0.95)
```

### Arguments

x — a ($k$ by 1) vector of $k$ groups' sample means for GRIMM or of each group's number of successful trials for BRIMM and PRIMM, where $k$ is the number of groups (or units) in a dataset.

| | |
|---|---|
| y | a (*k* by 1) vector composed of the standard errors of all groups for GRIMM or of each group's total number of trials for BRIMM and PRIMM. |
| covariates | (optional) a (*k* by *t*) matrix of covariates without a column of ones for an intercept term, where<br>*k* is the number of groups (or units) in a dataset and<br>*t* is the number of covariates (t $\geq$ 1). |
| 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. "gr" for Gaussian data, "br" for Binomial, and "pr" for Poisson. Default is "gr" |
| 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. |

## Details

gbp fits a Bayesian hierarchical model using generalized Stein's harmonic prior which enables good frequentist repeated sampling properties. The first-level is a distribution of observed data (likelihood) 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 multilevel model ($\sigma_j^2$ is assumed to be known and subscript *j* indicates *j*-th group (or unit) in a dataset):

$$(y_j \mid \theta_j) \ \sim \ indep\, N(\theta_j, \sigma_j^2)$$
$$(\theta_j \mid A,\ \mu_{0j}) \ \sim \ indep\, N(\mu_{0j},\ A)$$
$$\mu_{0j} \ = \ x_j'\beta$$

for $j = 1, \ldots, k$.

For Poisson data, gbp builds a two-level Poisson-Gamma multilevel model (a square bracket below indicates [mean, variance] of distribution):

$$(z_j \mid \theta_j) \ \sim \ indep\, Pois(n_j\theta_j)$$
$$(\theta_j \mid r,\ \mu_{0j}) \ \sim \ indep\, Gam(r\mu_{0j}, scale = 1/r) \ \sim \ indep\, Gam[\mu_{0j}, \mu_{0j}/r]$$
$$log(\mu_{0j}) \ = \ x_j'\beta$$

for $j = 1, \ldots, k$.

For Binomial data, gbp sets a two-level Binomial-Beta multilevel model:

$$(z_j \mid \theta_j) \ \sim \ indep\, Bin(n_j, \theta_j)$$
$$(\theta_j \mid r, \mu_{0j}) \ \sim \ indep\, Beta(r\mu_{0j},\ r(1-\mu_{0j})) \ \sim \ indep\, Beta[\mu_{0j},\ \mu_{0j}(1-\mu_{0j})\ /\ (r+1)]$$
$$logit(\mu_{0j}) \ = \ x_j'\beta$$

for $j = 1, \ldots, k$.

Theoretically, generalized Stein's harmonic prior is Uniform on the second level variance component (variance of the prior distribution), *i.e.*, *dA* for GRIMM and *d(1 / r)* (= $\frac{dr}{r^2}$) for BRIMM and PRIMM, leading to proper posterior distributions.

Under this setting, the argument `x` in gbp is a (*k* by 1) vector of *k* groups' sample means (*y* used in the Normal-Normal model above) for GRIMM or of each group's number of successful trials (*z*) for BRIMM and PRIMM, where *k* is the number of groups (or units) in a dataset.

The argument `y` (not $y$ used in the Normal-Normal model above) in `gbp` is a ($k$ by 1) vector composed of the standard errors ($\sigma$) of all groups for GRIMM or of each group's total number of trials ($n$) for BRIMM and PRIMM.

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 ($\mu_0$) a priori. In this case, assigning none of two optional arguments, such as "`gbp(z, n, model = "br")`", 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 some covariates (a $k$ by $t$ matrix, where $t \geq 1$) and do not know a mean of the prior distribution ($\mu_0$) a priori. In this case, assigning a $k$ by $t$ matrix (each column corresponds to one covariate), $X$, such as "`gbp(z, n, X, model = "pr")`", 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 ($\mu_0$) a priori. Now, we do not need to estimate regression coefficients at all because we know a true value of $\mu_0$ (strong assumption). Designating this value into the argument of `gbp` like "`gbp(y, se, mean.PriorDist = 3, model = "gr")`" 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`.

When it comes to estimating hyper-parameters, ($A$ or $r$) and $\beta$, `gbp` uses a mixture model with $\theta$ integrated out, first searching for a value that maximizes the marginal posterior distribution of $\alpha$ (= log($A$) for GRIMM or -log($r$) for BRIMM and PRIMM), and then looking for an estimate of $\beta$ (possibly a vector) that maximizes its posterior distribution (Likelihood) given previously estimated $\alpha$. `optim` is used for maximizing marginal and conditional posterior distributions.

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

## Value

An object of class `gbp` comprises of:

| | |
|---|---|
| `sample.mean` | sample mean of each group |
| `se` | if GRIMM, standard error of each group |
| `n` | if BRIMM and PRIMM, total number of trials of each group |
| `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 |
| `sd.shrinkage` | standard deviation of shrinkage estimate |
| `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 |
| `post.intv.upp` | upper bound of 100*Alpha% posterior interval |
| `model` | "gr" for GRIMM, "br" for BRIMM, and "pr" for PRIMM |
| `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` | alpha estimate |
| `a.var` | variance of alpha estimate |

### Author(s)

Joseph Kelly, Carl Morris, and Hyungsuk Tak

### References

Morris, C. and Tang, R. (2011). Estimating Random Effects via Adjustment for Density Maximization. *Statistical Science*. **26**. 271-287.

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

### Examples

```
# Loading datasets
data(schools)

# 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)
x1 <- rep(c(-1, 0, 1), 6)

y <- schools$y
se <- schools$se
x2 <- rep(c(-1, 0, 1, 2), 2)

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

#########
# GRIMM #
#########

g <- gbp(y, se)

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

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

#########
# BRIMM #
#########

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

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

### when we want to simulate pseudo datasets based on different values of r and
```

```
### of a regression coefficient (intercept),
### not using estimated values as true ones.
coverage(b, A.or.r = 60, reg.coef = -1, nsim = 10)

#########
# PRIMM #
#########

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

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

### when we want to simulate pseudo datasets based on different values of r and
### of a regression coefficient (intercept),
### not using estimated values as true ones.
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, #
##################################################################################

#########
# GRIMM #
#########

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

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

### when we want to simulate pseudo datasets based on different values of A,
### of regression coefficients, and of covariate, not using estimated values
### as true ones.
coverage(g, A.or.r = 9, reg.coef = c(10, 1), covariates = x2, nsim = 10)
### two values of reg.coef are for beta0 and beta1

#########
# BRIMM #
#########

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

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

### when we want to simulate pseudo datasets based on different values of r,
### of regression coefficients, and of covariate, not using estimated values
### as true ones.
coverage(b, A.or.r = 60, reg.coef = c(-1, 0), covariates = x1, nsim = 10)
### two values of reg.coef are for beta0 and beta1

#########
# PRIMM #
```

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

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

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

### when we want to simulate pseudo datasets based on different values of r,
### of regression coefficients, and of covariate, not using estimated values
### as true ones.
coverage(p, A.or.r = 60, reg.coef = c(-2, 0), covariates = x1, nsim = 10)
### two values of reg.coef are for beta0 and beta1

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

#########
# GRIMM #
#########

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

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

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

#########
# BRIMM #
#########

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

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

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

#########
# PRIMM #
#########

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

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

### when we want to simulate pseudo datasets based on different values of r and
```

```
### of 2nd level mean as true ones, not using estimated values as true ones.
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 data set 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)

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. #
#####################################################################################

#########
# GRIMM #
#########

g <- gbp(y, se)
```

```
g
summary(g)
plot(g)

#########
# BRIMM #
#########

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

#########
# PRIMM #
#########

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

---

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, ...)
```

## Arguments

| | |
|---|---|
| x | a resultant object of gbp function. |
| ... | 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 = "br")", 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 GRIMM, Gamma for PRIMM, and Beta for BRIMM). 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).

## Value

Two plots described in *details* will be displayed.

## 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. #
###################################################################################

#########
# GRIMM #
#########

g <- gbp(y, se)
plot(g)

#########
# BRIMM #
#########

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

#########
# PRIMM #
#########

p <- gbp(z, n, model = "pr")
plot(p)
```

print.gbp *Displying "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, ...)
```

## Arguments

| | |
|---|---|
| x | a resultant object of gbp function. |
| ... | 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 = "br")", the argument x is supposed to be b.

Users do not need to type "print(b)" but "b" itself is enough to call print.gbp.

## Value

print(gbp.object) will display

| | |
|---|---|
| sample.mean | sample mean of each group |
| se | if GRIMM, standard error of each group |
| n | if BRIMM and PRIMM, 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 |
| sd.shrinkage | standard deviation of shrinkage estimate |
| post.intv.low | lower bound of 100*Alpha% posterior interval |
| post.mean | posterior mean of each group |
| post.intv.upp | upper bound of 100*Alpha% posterior interval |
| post.sd | posterior standard deviation of each group |

## 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. #
###################################################################################

#########
# GRIMM #
#########

g <- gbp(y, se)
g

#########
# BRIMM #
#########

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

#########
# PRIMM #
#########

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

---

`print.summary.gbp` *Displying "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 GRIMM (or the same total number of trials for BRIMM and PRIMM). If they are not the same, `Main Summary` lists groups (units) with minimun, median, and maximum values of the standard error for GRIMM (or of the total number of trials for BRIMM and PRIMM). 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 GRIMM (or the same total number of trials for BRIMM and PRIMM), `Main Summary` lists groups (units) with minimun, 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`. To be specific, it shows estimate of $\alpha$ defined as log($A$) for GRIMM or -log($r$) for BRIMM and PRIMM and its standard deviation. It is actually a posterior mode.

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 case, `gbp.object` includes a regression result, `summary(gbp.object)` will display the result of resgression fit.

## Value

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

`Main summary`
: **Group w/ min(se or n)** an estimation result of a group (unit) with the minimum standard error for GRIMM or the minimum total number of trials for BRIMM and PRIMM.
: **Group w/ min(sample.mean)** appears instead of `Group w/ min(se or n)` when all the groups (units) have the same standard error for GRIMM or the same total number of trials for BRIMM and PRIMM.
: **Group w/ median(se or n)** an estimation result of group(s) (unit(s)) with the median standard error for GRIMM or the median total number of trials for BRIMM and PRIMM.
: **Group w/ median(sample.mean)** appears instead of `Group w/ median(se or n)` when all the groups (units) have the same standard error for GRIMM or the same total number of trials for BRIMM and PRIMM.
: **Group w/ max(se or n)** an estimation result of a group (unit) with the meximum standard error for GRIMM or the maximum total number of trials for BRIMM and PRIMM.
: **Group w/ max(sample.mean)** appears instead of `Group w/ max(se or n)` when all the groups (units) have the same standard error for GRIMM or the same total number of trials for BRIMM and PRIMM.
: **Mean over all groups** the overall average for all the groups (units) within each column.

`Second-level Variance Component Estimation Summary`
: **alpha.hat** a posterior mode of $\alpha$ defined as log($A$) for GRIMM or -log($r$) for BRIMM and PRIMM.
: **alpha.hat.sd** standard deviation of alpha.hat.

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, 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. #
###################################################################################

#########
# GRIMM #
#########

g <- gbp(y, se)
summary(g)

#########
# BRIMM #
#########

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

#########
# PRIMM #
#########

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

---

Rgbp *Fitting Bayesian Hierarchical Models*

---

## Description

Bayesian Hierarchical modeling for Gaussian (GRIMM), Binomial (BRIMM) and Poisson (PRIMM) data using generalized Stein's harmonic prior for good frequentist repeated sampling property.

## Details

| | |
|---|---|
| Package: | Rgbp |
| Type: | Package |
| Version: | 1.0.0 |
| Date: | 2013-03-16 |
| License: | GPL-2 |

## Author(s)

Joseph Kelly, Carl Morris, and Hyungsuk Tak

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

## References

Morris, C. and Tang, R. (2011). Estimating Random Effects via Adjustment for Density Maximization. *Statistical Science*. **26**. 271-287.

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

## Examples

```
# Loading datasets
data(schools)

# 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)
x1 <- rep(c(-1, 0, 1), 6)

y <- schools$y
se <- schools$se
x2 <- rep(c(-1, 0, 1, 2), 2)

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

#########
# GRIMM #
#########

g <- gbp(y, se)

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(g, nsim = 10)

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

```
#########
# BRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(b, nsim = 10)

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

#########
# PRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(p, nsim = 10)

### when we want to simulate psuedo datasets based on different values of r and
### of a regression coefficient (intercept),
### not using estimated values as true ones.
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, #
##################################################################################

#########
# GRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(g, nsim = 10)

### when we want to simulate psuedo datasets based on different values of A,
### of regression coefficients, and of covariate, not using estimated values
### as true ones.
coverage(g, A.or.r = 9, reg.coef = c(10, 1), covariates = x2, nsim = 10)
### two values of reg.coef are for beta0 and beta1

#########
# BRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
```

```
### as true ones.
coverage(b, nsim = 10)

### when we want to simulate psuedo datasets based on different values of r,
### of regression coefficients, and of covariate, not using estimated values
### as true ones.
coverage(b, A.or.r = 60, reg.coef = c(-1, 0), covariates = x1, nsim = 10)
### two values of reg.coef are for beta0 and beta1

#########
# PRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(p, nsim = 10)

### when we want to simulate psuedo datasets based on different values of r,
### of regression coefficients, and of covariate, not using estimated values
### as true ones.
coverage(p, A.or.r = 60, reg.coef = c(-2, 0), covariates = x1, nsim = 10)
### two values of reg.coef are for beta0 and beta1

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

#########
# GRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(g, nsim = 10)

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

#########
# BRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(b, nsim = 10)

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

```
#########
# PRIMM #
#########

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

### when we want to simulated psuedo datasets considering the estimated values
### as true ones.
coverage(p, nsim = 10)

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

---

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.

## References

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. #
###################################################################################

#########
# GRIMM #
#########

g <- gbp(y, se)
summary(g)

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

```
# BRIMM #
#########

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

#########
# PRIMM #
#########

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

# Index