Package 'binest'

September 22, 2026

Type Package

Title Estimation of Group Means and SDs from Binned Count Data

Version 0.3-1

Date 2026-09-22

Depends R (>= 3.5.0)

Imports splines, stats, utils

Suggests knitr, rmarkdown, R2jags

Description Education agencies often report school or district score distributions as the number of students scoring in each of several score ranges, or bins, separated by threshold scores, or cuts. The functions in the binest package translate those bin counts into estimates of the mean and standard deviation (SD). They do so using the heteroskedastic ordered probit (HETOP) model, which assumes that scores follow a normal distribution within each school or district, each of which has its own mean and SD. The binest package includes the fast_hetop() function, which fits the model much more quickly than previous implementations. The model is described by Reardon, Shear, Castellano and Ho (2017) <doi:10.3102/1076998616666279>; a Bayesian variant is described by Lockwood, Castellano and Shear (2018) <doi:10.3102/1076998618795124>.

License GPL (>= 2)

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Author Paul T. von Hippel [aut, cre],
David J. Hunter [aut],
J.R. Lockwood [aut] (Original HETOP package author)

Maintainer Paul T. von Hippel <ph3828@eid.utexas.edu>

Repository CRAN

Date/Publication 2026-09-22 20:10:43 UTC

Contents

`binest-package` *Estimation of Group Means and SDs from Binned Count Data*

Description

The three main package functions fit the same HETOP model, but differ dramatically in speed:

- `fast_hetop()` is the fastest. It can fit 1,000 schools or districts in less than 1 second. `fast_hetop(estimator = "ML")` produces maximum likelihood estimates and `fast_hetop(estimator = "EB_shrunk")` produces shrunken empirical Bayes estimates.
- `mle_hetop()` returns the same estimates as `fast_hetop(estimator = "ML")` but far more slowly. When given 1,000 districts, `mle_hetop()` fails to converge after an hour. Because of the speed difference, `mle_hetop()` is deprecated in favor of `fast_hetop(estimator = "ML")`.
- `fh_hetop()` returns estimates similar to `fast_hetop(estimator = "EB_shrunk")`, but far more slowly, taking over 40 minutes to estimate 1,000 districts. `fh_hetop()` fits the HETOP model using Markov Chain Monte Carlo, placing a Fay-Herriot prior over the group parameters and reporting Bayesian posterior means and SDs (Lockwood, Castellano and Shear 2018). Because `fast_hetop(estimator = "EB_shrunk")` can produce similar estimates more quickly, we have deprecated `fh_hetop()`.

The `mle_hetop()` and `fh_hetop()` functions are forked from the `HETOP` package by J. R. Lockwood, which was last updated in 2019 and archived on 2025-03-24 because email to the maintainer was undeliverable. When forking `mle_hetop()` and `fh_hetop()`, we removed two arguments (`fixedcuts` and `svals`) that some users found confusing. `mle_hetop()` and `fh_hetop()` are included in **binest** for comparison purposes (see `vignette("binest")`), but they are deprecated and will not be maintained.

Bundled data

The package ships with `tx_g6_math_2018`, which provides bin counts and mean scores for every Texas district that participated in the 2017-18 administration of the State of Texas Assessments of Academic Readiness (STAAR) Grade 6 mathematics test.

Author(s)

Paul T. von Hippel <ph3828@eid.utexas.edu>, David J. Hunter, and J. R. Lockwood.

References

Lockwood, J. R., Castellano, K. E., and Shear, B. R. (2018). Flexible Bayesian models for inferences from coarsened, group-level achievement data. *Journal of Educational and Behavioral Statistics*, 43(6), 663-692. doi:10.3102/1076998618795124

Reardon, S. F., Shear, B. R., Castellano, K. E., and Ho, A. D. (2017). Using heteroskedastic ordered probit models to recover moments of continuous test score distributions from coarsened data. *Journal of Educational and Behavioral Statistics*, 42(1), 3-45. doi:10.3102/1076998616666279

`fast_hetop` *fast_hetop(): Fast Estimation of the HETOP model for Binned Test Scores*

Description

`fast_hetop()` is the preferred function in this package. It can produce estimates for hundreds or thousands of schools or districts in less than a second. It can provide either maximum likelihood estimates (`estimator = "ML"`) or shrunken empirical Bayes estimates (`estimator = "EB_shrunk"`). It also has several features that the other package functions lack, including

- a test for whether scores really follow a normal distribution, as the HETOP model assumes;
- the ability to accept published cut scores;
- the ability to adjust standard errors downward when the data represent a population rather than a sample.

`fast_hetop(estimator = "ML")` provides identical estimates to `mle_hetop()`, but runs much more quickly, especially when the number of schools or districts is large.

`fast_hetop(estimator = "EB_shrunk")` provides similar but not identical estimates to `fh_hetop()`, and again runs much more quickly.

Usage

```
fast_hetop(ngk, cutpoints_known = FALSE, cutpoints = NULL,
          pooled_mean = 0, pooled_sd = 1,
          scope,
          estimator = "ML",
          tol = 1e-4, maxit = 100,
          conf.level = 0.95,
          estimate_unidentified_districts = TRUE)
```

Arguments

ngk	Data giving bin counts. Column k of row g counts how many students from school or district g scored in bin k.
cutpoints_known	Whether the cut scores are known. TRUE means they are known and supplied through the cutpoints argument. FALSE means they are unknown and must be estimated, assuming that the distribution pooled across all schools or districts has the mean and SD given by pooled_mean and pooled_sd.
cutpoints	A vector of cut scores. Required when cutpoints_known = TRUE. Cut scores must be listed in ascending order, and there must be one fewer cut score than the number of bins.
pooled_mean, pooled_sd	Used only when cutpoints_known = FALSE: the mean and SD to give the score distribution pooled across schools or districts. Defaults 0 and 1 report results on a standardized scale.
scope	Whether the data represent a sample of students or the whole population. Required, with no default, because it determines what the reported standard errors mean: "sample" counts both sampling and binning error, while "population" counts binning error alone.
estimator	Which estimator to compute. "ML" (default) returns maximum likelihood estimates for each school or district. "EB_shrunk" returns empirical Bayes estimates, which shrink each school or district toward the average, by an amount that depends on how precisely it is estimated. Shrinkage requires scope = "sample".
tol	Stop iterating when the mean and SD estimates change by less than this fraction of an SD. Default 1e-4.
maxit	Maximum number of iterations. Default 100.
conf.level	Confidence level for reported confidence intervals. Default 0.95.
estimate_unidentified_districts	If TRUE (default), districts with fewer than three populated bins will receive estimates. The HETOP model cannot identify both the mean and the SD of a district with fewer than three populated bins, but the mean can be estimated by assuming a typical value for the SD. Specifically, the SD is assumed to be the geometric mean of the SDs of the other districts.

Value

A list with the following components:

est_raw	Returned only when cutpoints_known = TRUE. Lists each school or district's mean, sd, mean_se, sd_se, mean_ci_lower, mean_ci_upper, sd_ci_lower, sd_ci_upper, along with scope, estimator, conf.level, cutpoints, and icc (the intracluster correlation).
est_std	Estimates on the standardized scale, where the population-weighted state mean is 0 and the total (within plus between) state SD is 1. Same elements as est_raw.

That total SD is computed from the observed between-group variance of the fitted means. Some implementations, Stata's `hetop.ado` among them, first subtract the estimation error in those means, and also apply a small-sample adjustment to the within-group component, so their standardized scale differs from this one by a small constant factor. Neither convention is more correct than the other, and quantities that do not depend on the scale (correlations, ratios of SDs, cut spacing ratios) are unaffected.

`gof` For each school or district, a Pearson chi-square testing the goodness of fit (gof) of the normal distribution that the HETOP model assumes. A data frame with columns `chisq`, `df`, `p`, and `min_exp` (the smallest expected count, for callers who wish to screen on it). Degrees of freedom are `K - 3`, where `K` is the number of bins, so at least four bins are needed for the test to have anything to test. The power of the test is low unless the school or district is large, so a non-significant p value does not establish that scores are normally distributed.

`iter_info` Diagnostics from the fit, useful for checking that it behaved. Not needed for ordinary use. It records

- how the per-group iteration went: `within_iterations` (total passes), `within_iterations_per_dis` `within_converged`, `within_exact3` (groups solved in closed form because exactly three bins were populated), and `within_unidentified`;
- when the cut scores were estimated, how that outer loop went: `iterated`, `iterations`, `converged`, and the two tolerances used, `tol` and `tol_cutpoints_std`;
- how any underidentified groups were salvaged: `salvage_counts`, the individual counts `salvaged_two_bin`, `salvaged_one_interior`, `salvaged_one_bottom` and `salvaged_one_top`, and the borrowed values `mu_pool` and `sigma_pool`;
- under `estimator = "EB_shrunk"`, the shrinkage quantities: `eb_shrink`, `eb_weights`, `eb_tau2` and `eb_tau2_mu` (between-group variance of the log-SDs and of the means), and the shrinkage targets `eb_log_mean` and `eb_mean_mu`.

Author(s)

Paul T. von Hippel and David J. Hunter.

References

Reardon S., Shear B.R., Castellano K.E. and Ho A.D. (2017). "Using heteroskedastic ordered probit models to recover moments of continuous test score distributions from coarsened data," *Journal of Educational and Behavioral Statistics* 42(1):3–45.

Lockwood J.R., Castellano K.E. and Shear B.R. (2018). "Flexible Bayesian models for inferences from coarsened, group-level achievement data," *Journal of Educational and Behavioral Statistics.* 43(6):663–692.

Examples

```
set.seed(1001)
G   <- 10

## Let means and SDs vary across the groups.
```

```
mug       <- seq(from = -2.0, to = 2.0, length = G)
sigmag    <- seq(from =  2.0, to = 0.8, length = G)
cutpoints <- c(-1.0, 0.0, 0.8)
ng        <- rep(1000, G)
ngk       <- gendata_hetop(G, K = 4, ng, mug, sigmag, cutpoints)

## Here the counts were simulated by sampling ng scores per group, so
## scope = "sample" and the reported SEs will include sampling as well
## as binning error.
##
## Cutpoints known: both est_raw (test-score scale) and est_std
## (standardized scale) are returned.
bm <- fast_hetop(ngk, cutpoints_known = TRUE, cutpoints = cutpoints,
                 scope = "sample")
print(cbind(true = mug,       est = bm$est_raw$mean))
print(cbind(true = sigmag,    est = bm$est_raw$sd))
print(cbind(true = cutpoints, est = bm$est_raw$cutpoints))
print(cbind(est = bm$est_raw$mean,
            se  = bm$est_raw$mean_se,
            lo  = bm$est_raw$mean_ci_lower,
            hi  = bm$est_raw$mean_ci_upper))

## If the data represented the whole population, then scope =
## "population", and the reported standard errors are smaller because
## they reflect only binning error.
bm_pop <- fast_hetop(ngk, cutpoints_known = TRUE, cutpoints = cutpoints,
                     scope = "population")
print(cbind(sample_se     = bm$est_raw$mean_se,
            population_se = bm_pop$est_raw$mean_se))

## If cutpoints_known = FALSE, then cutpoints are unknown and are
## estimated on a scale with mean and SD given by pooled_mean
## (default 0) and pooled_sd (default 1).
bm2 <- fast_hetop(ngk, scope = "sample")
print(bm2$est_std$cutpoints)
print(bm2$est_std$mean)

## If the cut scores are unknown but the pooled mean and SD are known,
## you can pass the pooled mean and SD and the estimates will come back
## on that scale.
bm3 <- fast_hetop(ngk, pooled_mean = 1640.2, pooled_sd = 138.6,
                  scope = "sample")
print(bm3$est_std$cutpoints)
```

---

fh_hetop *fh_hetop(): Bayesian Estimation of the HETOP Model using JAGS*

Description

fh_hetop() calculates Bayesian estimates of the HETOP model parameters using Markov Chain Monte Carlo. It depends on R2jags::jags, and JAGS must be installed for fh_hetop() to run;

JAGS can be downloaded from https://sourceforge.net/projects/mcmc-jags/. fh_hetop() produces estimates on a standardized scale, not on the native scale of the assessment.

Unlike the other HETOP estimators in this package, fh_hetop() can accept covariates to predict district means and log standard deviations. All covariates must be centered so that they sum to zero across groups.

Further details on the fh_hetop() model are provided by Lockwood, Castellano and Shear (2018).

Usage

```
fh_hetop(ngk, p, m, gridL, gridU, Xm=NULL, Xs=NULL,
seed=12345, modelfileonly = FALSE, modloc=NULL, ...)
```

Arguments

ngk	Data giving bin counts. Column k of row g counts how many students from school or district g scored in bin k.
p	Vector of length 2 giving degrees of freedom for cubic spline basis to parameterize Efron priors for group means and group standard deviations; see References.
m	Vector of length 2 giving number of grid points to parameterize Efron priors for group means and group standard deviations; see References.
gridL	Vector of length 2 of lower bounds for grids to parameterize Efron priors for group means and group standard deviations; see References.
gridU	Vector of length 2 of upper bounds for grids to parameterize Efron priors for group means and group standard deviations; see References.
Xm	Optional matrix of covariates for the group means.
Xs	Optional matrix of covariates for the log group standard deviations.
seed	Passed to set.seed.
modelfileonly	If TRUE, function returns location of JAGS model file only, without running JAGS. Default is FALSE.
modloc	Optional character vector of length 1 providing the full path to the name of file where the JAGS model code will be written. Defaults to NULL, in which case the code will be written to a temporary file.
...	Additional arguments to R2jags::jags.

Value

A object of class rjags, with additional information specific to the FH-HETOP model. The additional information is stored as a list called fh_hetop_extras with the following components:

Finfo	A list containing information used to estimate the population distribution of the residuals from the FH-HETOP model. Posterior samples of the parameters defining the residual distribution can be found in the BUGSoutput element of the returned object.
Dinfo	A list containing information about the data used to the fit the model, including the counts, covariates and fixed cutpoints.

`waicinfo` A list containing information about the WAIC for the estimated model; see help file for `waic_hetop()`.

`est_star_samps` A list with posterior samples of parameters with respect to the 'star' scale which defines the location and scale of the group means and standard deviations that corresponds to a marginal population mean of zero and marginal population standard deviation of 1. Additional details in help file for `mle_hetop()`

`est_star_mug` A dataframe containing various estimates of the group means on the 'star' scale, including posterior means, Constrained Bayes and Triple-Goal estimates. Additional details in help file for `triple_goal()`.

`est_star_sigmag` A dataframe containing various estimates of the group standard deviations on the 'star' scale, including posterior means, Constrained Bayes and Triple-Goal estimates. Additional details in help file for `triple_goal()`.

Deprecated

`fh_hetop()` is deprecated in favor of `fast_hetop(estimator = "EB_shrunk")`, which produces similar estimates much faster and without external dependencies such as JAGS. If `fh_hetop()`'s runtime can be improved, we may revive it for users who would like to use covariates or prefer fully Bayesian estimation. For now, though, `fh_hetop()` is deprecated for slow performance.

Author(s)

J.R. Lockwood <jrlockwood@ets.org> (original implementation); Paul T. von Hippel <ph3828@eid.utexas.edu> (modifications for **binest**).

References

Efron B. (2016). "Empirical Bayes deconvolution estimates," *Biometrika* 103(1):1–20.

Lockwood J.R., Castellano K.E. and Shear B.R. (2018). "Flexible Bayesian models for inferences from coarsened, group-level achievement data," *Journal of Educational and Behavioral Statistics.* 43(6):663–692.

See Also

`R2jags::jags`

Examples

```
## Not run:
## fh_hetop() requires JAGS, an external system binary; see
## https://sourceforge.net/projects/mcmc-jags/.  The example below
## is wrapped in \dontrun{} so that it is not executed by R CMD
## check, but should run interactively once JAGS is installed.

set.seed(1001)

## define mean-centered covariates
G  <- 12
```

```
z1 <- sample(c(0,1), size=G, replace=TRUE)
z2 <- 0.5*z1 + rnorm(G)
Z  <- cbind(z1 - mean(z1), z2 = z2 - mean(z2))

## define true parameters dependent on covariates
beta_m    <- c(0.3,  0.8)
beta_s    <- c(0.1, -0.1)
mug       <- Z[,1]*beta_m[1] + Z[,2]*beta_m[2] + rnorm(G, sd=0.3)
sigmag    <- exp(0.3 + Z[,1]*beta_s[1] + Z[,2]*beta_s[2] + 0.2*rt(G, df=7))
cutpoints <- c(-1.0, 0.0, 1.2)

## generate data
ng   <- rep(200,G)
ngk  <- gendata_hetop(G, K = 4, ng, mug, sigmag, cutpoints)
print(ngk)

## fit FH-HETOP model including covariates
## NOTE: using an extremely small number of iterations for testing,
##       so that convergence is not expected
m <- fh_hetop(ngk, p = c(10,10),
              m = c(100, 100), gridL = c(-5.0, log(0.10)),
              gridU = c(5.0, log(5.0)), Xm = Z, Xs = Z,
              n.iter = 100, n.burnin = 50)

print(m)
print(names(m$fh_hetop_extras))

s <- m$BUGSoutput$summary
print(data.frame(truth = c(beta_m, beta_s), s[grep("beta", rownames(s)),]))

print(cor(mug,    s[grep("mu",    rownames(s)),"mean"]))
print(cor(sigmag, s[grep("sigma", rownames(s)),"mean"]))

## manual calculation of WAIC (see help file for waic_hetop)
tmp <- waic_hetop(ngk, m$BUGSoutput$sims.matrix)
identical(tmp, m$fh_hetop_extras$waicinfo)

## End(Not run)
```

---

gendata_hetop *gendata_hetop(): Generate count data from Heteroskedastic Ordered Probit (HETOP) Model*

Description

gendata_hetop() can simulate data that satisfies the assumptions of the HETOP model. It can be used to test the properties of an estimator when the HETOP assumptions are met. It generates ng scores for each of G groups (e.g., districts) and bins the scores into K bins. It requires the bin cutpoints and the mean and SD for each group.

Usage

```
gendata_hetop(G, K, ng, mug, sigmag, cutpoints)
```

Arguments

G	Number of groups.
K	Number of ordinal categories.
ng	Vector of length G providing the total number of units in each group.
mug	Vector of length G giving the latent variable mean for each group.
sigmag	Vector of length G giving the latent variable standard deviation for each group.
cutpoints	Vector of length (K-1) giving cutpoint locations, held constant across groups, that map the continuous latent variable to the observed categorical variable.

Details

For each group g, the function generates ng IID normal random variables with mean mug[g] and standard deviation sigmag[g], and then assigns each to one of K ordered groups, depending on cutpoints. The resulting data for a group is a table of category counts summing to ng[g].

Value

A G x K matrix where column k of row g provides the number of simulated units from group g falling into category k.

Author(s)

J.R. Lockwood <jrlockwood@ets.org>

References

Reardon S., Shear B.R., Castellano K.E. and Ho A.D. (2017). "Using heteroskedastic ordered probit models to recover moments of continuous test score distributions from coarsened data," *Journal of Educational and Behavioral Statistics* 42(1):3–45.

Lockwood J.R., Castellano K.E. and Shear B.R. (2018). "Flexible Bayesian models for inferences from coarsened, group-level achievement data," *Journal of Educational and Behavioral Statistics.* 43(6):663–692.

Examples

```
set.seed(1001)

## define true parameters
G         <- 10
mug       <- seq(from= -2.0, to= 2.0, length=G)
sigmag    <- seq(from=  2.0, to= 0.8, length=G)
cutpoints <- c(-1.0, 0.0, 0.8)

## generate data with large counts
```

```
ng   <- rep(100000,G)
ngk  <- gendata_hetop(G, K = 4, ng, mug, sigmag, cutpoints)
print(ngk)

## compare theoretical and empirical cell probabilities
phat  <- ngk / ng
ptrue <- t(sapply(1:G, function(g){
    tmp <- c(pnorm(cutpoints, mug[g], sigmag[g]), 1)
    c(tmp[1], diff(tmp))
}))
print(max(abs(phat - ptrue)))
```

`mle_hetop` *mle_hetop(): Joint Maximum Likelihood Estimation of the HETOP Model from Bin Counts*

Description

`mle_hetop()` returns maximum likelihood estimates from the HETOP model. Estimates are identical to those from `fast_hetop(estimator = "ML")`, but `mle_hetop()` runs much more slowly because it tries to estimate all districts at once and uses numerical approximations for the first and second derivatives.

This implementation is forked, with small changes, from the `HETOP` package by J. R. Lockwood.

Usage

```
mle_hetop(ngk, iterlim = 1500, ...)
```

Arguments

`ngk`	Data giving bin counts. Column `k` of row `g` counts how many students from school or district `g` scored in bin `k`.
`iterlim`	Maximum number of iterations used in optimization (passed to `nlm`).
`...`	Any other arguments for `nlm`.

Details

This function requires at least 3 populated bins to identify the model. For districts with fewer than 3 populated bins, we identify the mean by assuming that the SD is equal to the geometric mean of the SDs for other districts.

This function can report estimates on four different scales:

1. the original estimation scale with two fixed cutpoints;
2. a scale defined by forcing the group means and log group standard deviations each to have weighted mean of zero, where weights are proportional to the total count for each group;
3. a scale where the population mean of the latent variable is zero and the population standard deviation is one; and

4. a scale similar to (3) but where a bias correction is applied. See Reardon et al. (2017) for details on this bias correction.

The function also returns an estimated intracluster correlation (ICC) of the latent variable, defined as the ratio of the between-group variance of the latent variable to its marginal variance. Scales (1)-(3) above lead to the same estimated ICC; scale (4) uses a bias-corrected estimate of the ICC which will not in general equal the estimate from scales (1)-(3).

Value

A list with the following components:

`est_fc`	A list of estimated group means, group standard deviations, cutpoints and ICC on scale (1).
`est_zero`	A list of estimated group means, group standard deviations, cutpoints and ICC on scale (2).
`est_star`	A list of estimated group means, group standard deviations, cutpoints and ICC on scale (3).
`est_starbc`	A list of estimated group means, group standard deviations, cutpoints and ICC on scale (4).
`nlmdetails`	The object returned by `nlm` that summarizes detailed of the optimization.
`pstatus`	A dataframe, with one row for each group, summarizing the estimation status of the mean and standard deviation for each group. A value of `est` means that the parameter was estimated without constraints. A value of `mean`, used for the group standard deviations, indicates that the parameter was constrained. Values of `min` or `max`, used for the group means, indicate that the parameter was constrained.

Deprecated

`mle_hetop()` is deprecated in favor of `fast_hetop(estimator = "ML")`, which produces ML estimates much faster. `mle_hetop()` is preserved for comparison but will not be maintained.

Author(s)

J. R. Lockwood (original implementation); David J. Hunter and Paul T. von Hippel <ph3828@eid.utexas.edu> (modifications for **binest**).

References

Reardon S., Shear B.R., Castellano K.E. and Ho A.D. (2017). "Using heteroskedastic ordered probit models to recover moments of continuous test score distributions from coarsened data," *Journal of Educational and Behavioral Statistics* 42(1):3–45.

Lockwood J.R., Castellano K.E. and Shear B.R. (2018). "Flexible Bayesian models for inferences from coarsened, group-level achievement data," *Journal of Educational and Behavioral Statistics.* 43(6):663–692.

Examples

```
set.seed(1001)

## define true parameters
G         <- 10
mug       <- seq(from= -2.0, to= 2.0, length=G)
sigmag    <- seq(from=  2.0, to= 0.8, length=G)
cutpoints <- c(-1.0, 0.0, 0.8)

## generate data with large counts
ng   <- rep(100000,G)
ngk  <- gendata_hetop(G, K = 4, ng, mug, sigmag, cutpoints)
print(ngk)

## compute MLE and check parameter recovery (cutpoints derived from data):
m    <- mle_hetop(ngk)
print(cbind(true = mug,       est = m$est_fc$mug))
print(cbind(true = sigmag,    est = m$est_fc$sigmag))
print(cbind(true = cutpoints, est = m$est_fc$cutpoints))

## estimates on other scales:
p    <- ng/sum(ng)
print(sum(p * m$est_zero$mug))
print(sum(p * log(m$est_zero$sigmag)))

print(sum(p * m$est_star$mug))
print(sum(p * (m$est_star$mug^2 + m$est_star$sigmag^2)))

## dealing with sparse counts
ngk_sparse <- matrix(rpois(G*4, lambda=5), ncol=4)
ngk_sparse[1,] <- c(5,8,0,0)
ngk_sparse[2,] <- c(0,10,10,0)
ngk_sparse[3,] <- c(12,0,0,0)
ngk_sparse[4,] <- c(0,0,0,10)
print(ngk_sparse)

m    <- mle_hetop(ngk_sparse)
print(m$pstatus)
print(unique(m$est_fc$sigmag[1:4]))
print(exp(mean(log(m$est_fc$sigmag[5:10]))))
print(m$est_fc$mug[3])
print(min(m$est_fc$mug[-3]))
print(m$est_fc$mug[4])
print(max(m$est_fc$mug[-4]))
```

`triple_goal` *Shen and Louis (1998) Triple Goal Estimators*

Description

`triple_goal()` is a Bayesian helper function used by `fh_hetop()`. It implements the "Triple Goal" estimates of Shen and Louis (1998) for a vector of parameters given a sample from the posterior distribution of those parameters. Also computes "constrained Bayes" estimators of Ghosh (1992).

Usage

```
triple_goal(s, stop.if.ties = FALSE, quantile.type = 7)
```

Arguments

`s`	A (`n` x `K`) matrix of `n` samples of `K` group parameters with no missing values.
`stop.if.ties`	logical; if TRUE, function stops if any units have identical posterior mean ranks; otherwise breaks ties at random.
`quantile.type`	`type` argument to `quantile` function for different methods of computing quantiles.

Details

In typical applications, the matrix `s` will be a sample of size `n` from the joint posterior distribution of a vector of `K` group-specific parameters. Both the triple goal and constrained Bayes estimators are designed to mitigate problems arising from underdispersion of posterior means; see references.

Value

A dataframe with `K` rows with fields:

`theta_pm`	Posterior mean estimates of group parameters.
`theta_psd`	Posterior standard deviation estimates of group parameters.
`theta_cb`	"Constrained Bayes" estimates of group parameters using formula in Shen and Louis (1998).
`theta_gr`	"Triple Goal" estimates of group parameters using algorithm defined in Shen and Louis (1998).
`rbar`	Posterior means of ranks of group parameters (1=lowest).
`rhat`	Integer ranks of group parameters (=rank(rbar)).

Author(s)

J.R. Lockwood <jrlockwood@ets.org>

References

Shen W. and Louis T.A. (1998). "Triple-goal estimates in two-stage hierarchical models," *Journal of the Royal Statistical Society, Series B* 60(2):455-471.

Ghosh M. (1992). "Constrained Bayes estimation with applications," *Journal of the American Statistical Association* 87(418):533-540.

Examples

```
set.seed(1001)
.K <- 50
.nsamp <- 500
.theta_true <- rnorm(.K)
.s <- matrix(.theta_true, ncol=.K, nrow=.nsamp, byrow=TRUE) +
      matrix(rnorm(.K*.nsamp, sd=0.4), ncol=.K, nrow=.nsamp)
.e <- triple_goal(.s)
str(.e)
head(.e)
```

---

tx_g6_math_2018 *tx_g6_math_2018: Texas STAAR Grade 6 Mathematics Scores: District-Level Bin Counts*

Description

`tx_g6_math_2018` is a bundled dataset used in the vignette. It can be used to test the properties of HETOP estimators in empirical data that may violate some model assumptions.

`tx_g6_math_2018` provides district-level counts of students in each of four score bins on the Texas State of Texas Assessments of Academic Readiness (STAAR) Grade 6 mathematics test, 2017-18 administration. For each district the dataset also reports the average score reported by the Texas Education Agency, which can be used as ground truth for evaluating estimators that recover district means from binned counts.

Usage

```
data(tx_g6_math_2018)
```

Format

A data frame with 1151 rows and 8 columns:

district_id Sequential integer identifier (1 to 1151).

district_name District name (`character`).

n_tested Total students tested in the district.

unsatisfactory Students scoring below 1536 (proficiency category "Did Not Meet Grade Level").

approaches Students scoring in [1536, 1653) ("Approaches Grade Level").

meets Students scoring in [1653, 1772) ("Meets Grade Level").

masters Students scoring >= 1772 ("Masters Grade Level").

reported_mean District average score, computed by the Texas Education Agency from individual student scores.

Details

The three published cut scores defining the bin boundaries are 1536, 1653, and 1772. The administrative floor of the STAAR scale is 1062 and the ceiling is 2143. Of the 1151 districts, 1014 have nonzero counts in all four bins, 120 have nonzero counts in three bins, and 17 have nonzero counts in two bins.

Source

Texas Education Agency, Academic Performance Reports (TAPR), 2017-18. Compiled by D.\ J.\ Hunter and P.\ T.\ von Hippel.

Examples

```
data(tx_g6_math_2018)
str(tx_g6_math_2018)

## Recover district means using fast_hetop with known cutpoints.
ngk <- with(tx_g6_math_2018,
            cbind(unsatisfactory, approaches, meets, masters))
## These are administrative counts covering every tested student, so
## each district's students are its whole population: scope = "population".
fit <- fast_hetop(ngk, cutpoints_known = TRUE,
                  cutpoints = c(1536, 1653, 1772),
                  scope = "population")

## Correlation with reported truth on the test-score scale.
## (The 17 districts with fewer than three populated bins cannot
## identify both a mean and an SD; by default fast_hetop() estimates
## their means using a borrowed SD.  Pass
## estimate_unidentified_districts = FALSE to get NA instead, in which
## case use = "complete.obs" is needed here.)
cor(fit$est_raw$mean, tx_g6_math_2018$reported_mean)
```

`waic_hetop` *WAIC for FH-HETOP model*

Description

`waic_hetop()` is a helper function used by `fh_hetop()`. `waic_hetop()` computes the Watanabe-Akaike information criterion (WAIC) for the FH-HETOP model using the data and posterior samples of the group means, group standard deviations and cutpoints.

Usage

```
waic_hetop(ngk, samps)
```

Arguments

ngk	Data giving bin counts. Column k of row g counts how many students from school or district g scored in bin k.
samps	A matrix of posterior samples that includes at least the group means, group standard deviations and the cutpoints. Column names for these three collections of parameters must contain the strings 'mu', 'sigma' and 'cuts', respectively.

Details

Although this function can be called directly by the user, it is primarily intended to be used to compute WAIC as part of the function `fh_hetop()`. Details on the WAIC calculation are provided by Vehtari and Gelman (2017).

Value

A list with the following components:

lpd_hat	Part 1 of the WAIC calculation: the estimated log pointwise predictive density, summed across groups.
phat_waic	Part 2 of the WAIC calculation: the effective number of parameters.
waic	The WAIC criterion: -2 times (lpd_hat - phat_waic).

Author(s)

J.R. Lockwood <jrlockwood@ets.org>

References

Lockwood J.R., Castellano K.E. and Shear B.R. (2018). "Flexible Bayesian models for inferences from coarsened, group-level achievement data," *Journal of Educational and Behavioral Statistics.* 43(6):663–692.

Vehtari A., Gelman A. and Gabry J. (2017). "Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC," *Statistics and Computing.* 27(5):1413–1432.

Examples

```
if (requireNamespace("R2jags", quietly = TRUE)) {
  set.seed(42)
  G <- 10
  ngk <- gendata_hetop(G = G, K = 4, ng = rep(50, G),
                       mug = rnorm(G), sigmag = exp(rnorm(G, 0, 0.2)),
                       cutpoints = c(-1, 0, 1))
  m <- fh_hetop(ngk,
                p = c(10, 10), m = c(100, 100),
                gridL = c(-5, log(0.10)), gridU = c(5, log(5.0)),
                n.iter = 200, n.burnin = 100, seed = 1)
  waic <- waic_hetop(ngk, m$BUGSoutput$sims.matrix)
  print(waic)
}
```

Index