

# Package ‘shrinkem’

May 9, 2026

**Type** Package

**Title** Approximate Bayesian Regularization for Parsimonious Estimates

**Date** 2024-10-01

**Version** 0.2.0

**Author** Joris Mulder [aut, cre],  
Diana Karimova [aut, ctb],  
Sara van Erp [ctb]

**Maintainer** Joris Mulder <j.mulder3@tilburguniversity.edu>

**Description** Approximate Bayesian regularization using Gaussian approximations. The input is a vector of estimates and a Gaussian error covariance matrix of the key parameters. Bayesian shrinkage is then applied to obtain parsimonious solutions. The method is described on Karimova, van Erp, Leenders, and Mulder (2024) <[DOI:10.31234/osf.io/2g8qm](https://doi.org/10.31234/osf.io/2g8qm)>. Gibbs samplers are used for model fitting. The shrinkage priors that are supported are Gaussian (ridge) priors, Laplace (lasso) priors (Park and Casella, 2008 <[DOI:10.1198/016214508000000337](https://doi.org/10.1198/016214508000000337)>), and horse-shoe priors (Carvalho, et al., 2010; <[DOI:10.1093/biomet/asq017](https://doi.org/10.1093/biomet/asq017)>). These priors include an option for grouped regularization of different subsets of parameters (Meier et al., 2008; <[DOI:10.1111/j.1467-9868.2007.00627.x](https://doi.org/10.1111/j.1467-9868.2007.00627.x)>). F priors are used for the penalty parameters  $\lambda^2$  (Mulder and Pericchi, 2018 <[DOI:10.1214/17-BA1092](https://doi.org/10.1214/17-BA1092)>). This correspond to half-Cauchy priors on  $\lambda$  (Carvalho, Polson, Scott, 2010 <[DOI:10.1093/biomet/asq017](https://doi.org/10.1093/biomet/asq017)>).

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** stats, mvtnorm, extraDistr, brms, CholWishart, matrixcalc

**Suggests** testthat

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2024-10-05 10:20:03 UTC

## Contents

F . . . . .	2
mvF . . . . .	3
shrinkem . . . . .	4
<b>Index</b>	<b>7</b>

---

### F *The (scaled) F Distribution*

---

#### Description

Density and random generation for the F distribution with first degrees of freedom  $df1$ , second degrees of freedom  $df2$ , and scale parameter  $\beta$ .

#### Usage

```
dF(x, df1, df2, beta, log = FALSE)
```

```
rF(n, df1, df2, beta)
```

#### Arguments

x	vector of quantities.
df1	First degrees of freedom
df2	Second degrees of freedom
beta	Scale parameter
log	logical; if TRUE, density is given as $\log(p)$ .
n	number of draws

#### Value

`dF` gives the probability density of the F distribution. `rF` gives random draws from the F distribution.

#### References

Mulder and Pericchi (2018). The Matrix-F Prior for Estimating and Testing Covariance Matrices. *Bayesian Analysis*, 13(4), 1193-1214. <<https://doi.org/10.1214/17-BA1092>>

#### Examples

```
draws_F <- rF(n=1e4, df1=2, df2=4, beta=1)
hist(draws_F, 500, xlim=c(0, 10), freq=FALSE)
seqx <- seq(0, 10, length=1e5)
lines(seqx, dF(seqx, df1=2, df2=4, beta=1), col=2, lwd=2)
```

**Description**

Density and random generation for the matrix variate F distribution with first degrees of freedom df1, second degrees of freedom df2, and scale matrix B.

**Usage**

```
dmvF(x, df1, df2, B, log = FALSE)
```

```
rmvF(n, df1, df2, B)
```

**Arguments**

x	Positive definite matrix of quantities.
df1	First degrees of freedom
df2	Second degrees of freedom
B	Positive definite scale matrix
log	logical; if TRUE, density is given as log(p).
n	Number of draws

**Value**

dmvF returns the probability density of the matrix F distribution. rmvF returns a numeric array, say R, of dimension  $p \times p \times n$ , where each element  $R[, , i]$  is a positive definite matrix, a realization of the matrix F distribution.

**References**

Mulder and Pericchi (2018). The Matrix-F Prior for Estimating and Testing Covariance Matrices. Bayesian Analysis, 13(4), 1193-1214. <<https://doi.org/10.1214/17-BA1092>>

**Examples**

```
set.seed(20180222)
draws_F <- rmvF(n=1, df1=2, df2=4, B=diag(2))
dmvF(draws_F[, , 1], df1=2, df2=4, B=diag(2))
```

shrinkem

*Fast Bayesian regularization using Gaussian approximations***Description**

The shrinkem function can be used for regularizing a vector of estimates using Bayesian shrinkage methods where the uncertainty of the estimates are assumed to follow a Gaussian distribution.

**Usage**

```
shrinkem(
  x,
  Sigma,
  type,
  group,
  iterations,
  burnin,
  store,
  cred.level,
  df1,
  df2,
  scale2,
  lambda2.fixed,
  lambda2,
  ...
)
```

**Arguments**

x	A vector of estimates.
Sigma	A covariance matrix capturing the uncertainty of the estimates (e.g., error covariance matrix).
type	A character string which specifies the type of regularization method is used. Currently, the types "ridge", "lasso", and "horseshoe", are supported.
group	A vector of integers denoting the group membership of the estimates, where each group receives a different global shrinkage parameter which is adapted to the observed data.
iterations	Number of posterior draws after burnin. Default = 5e4.
burnin	Number of posterior draws in burnin. Default = 1e3.
store	Store every store-th draw from posterior. Default = 1 (implying that every draw is stored).
cred.level	The significance level that is used to check whether a parameter is nonzero depending on whether 0 is contained in the credible interval. The default is cred.level = 0.95.

df1	First hyperparameter (degrees of freedom) of the prior for a shrinkage parameter $\lambda^2$ , which follows a $F(df1, df2, scale2)$ distribution. The default is $df1 = 1$ . For $df1 = 1$ , this corresponds to half-t distribution for $\lambda$ with degrees of freedom $df2$ and scale parameter $\sqrt{scale2/df2}$ .
df2	Second hyperparameter (degrees of freedom) of the prior for a shrinkage parameter $\lambda^2$ , which follows a $F(df1, df2, scale2)$ distribution. The default is $df2 = 1$ .
scale2	Second hyperparameter (scale parameter) of the prior for a shrinkage parameter $\lambda^2$ , which follows a $F(df1, df2, scale2)$ distribution. The default is $df2 = 1e3$ .
lambda2.fixed	Logical indicating whether the penalty parameters(s) is/are fixed. Default is FALSE.
lambda2	Positive scalars of length equal to the number of groups in 'group'. The argument is only used if the argument 'lambda2.fixed' is 'TRUE'.
...	Parameters passed to and from other functions.

### Value

The output is an object of class `shrinkem`. The object has elements:

- `estimates`: A data frame with the input estimates, the shrunken posterior mean, median, and mode, the lower and upperbound of the credibility interval based on the shrunken posterior, and a logical which indicates if zero is contained in the credibility interval.
- `draws`: List containing the posterior draws of the effects (`beta`), the prior parameters (`tau2`, `gamma2`), and the penalty parameters (`psi2` and `lambda2`).
- `dim.est`: The dimension of the input estimates of `beta`.
- `input.est`: The input vector of the unshrunken estimates of `beta`.
- `call`: Input call.

### References

Karimovo, van Erp, Leenders, and Mulder (2024). Honey, I Shrunk the Irrelevant Effects! Simple and Fast Approximate Bayesian Regularization. <<https://doi.org/10.31234/osf.io/2g8qm>>

### Examples

```
# EXAMPLE
estimates <- -5:5
covmatrix <- diag(11)
# Bayesian horseshoe where all beta's have the same global shrinkage
# (using default 'group' argument)
shrink1 <- shrinkem(estimates, covmatrix, type="horseshoe")
# posterior modes of middle three estimates are practically zero

# plot posterior densities
old.par.mfrow <- par(mfrow = c(1,1))
old.par.mar <- par(mar = c(0, 0, 0, 0))
par(mfrow = c(11,1))
```

```
par(mar = c(1,2,1,2))
for(p in 1:ncol(shrink1$draws$beta)){plot(density(shrink1$draws$beta[,p]),
  xlim=c(-10,10),main=colnames(shrink1$draws$beta)[p])}
par(mfrow = old.par.mfrow)
par(mar = old.par.mar)

# Bayesian horseshoe where first three and last three beta's have different
# global shrinkage parameter than other beta's
shrink2 <- shrinkem(estimates, covmatrix, type="horseshoe",
  group=c(rep(1,3),rep(2,5),rep(1,3)))
# posterior modes of middle five estimates are virtually zero

# plot posterior densities
par(mfrow = c(11,1))
par(mar = c(1,2,1,2))
for(p in 1:ncol(shrink2$draws$beta)){plot(density(shrink2$draws$beta[,p]),xlim=c(-10,10),
  main=colnames(shrink2$draws$beta)[p])}
par(mfrow = old.par.mfrow)
par(mar = old.par.mar)
```

# Index

dF (F), 2  
dmvF (mvF), 3  
F, 2  
mvF, 3  
rF (F), 2  
rmvF (mvF), 3  
shrinkem, 4