Package ‘BHSBVAR’

January 23, 2019

Encoding UTF-8

Type Package

Title Structural Bayesian Vector Autoregression Models

Version 1.0.1

Date 2019-01-23

Author Paul Richardson

Maintainer Paul Richardson <p.richardson.54391@gmail.com>

Description Provides a function for running Structural Bayesian Vector Autoregression models with the method developed by Baumeister and Hamilton (2015) <doi:10.3982/ECTA12356>, Baumeister and Hamilton (2017) <doi:10.3386/w24167>, and Baumeister and Hamilton (2018) <doi:10.1016/j.jmoneco.2018.06.005>. Functions for plotting impulse responses, historical decompositions, and posterior distributions of model parameters are also provided.

License GPL (>= 3)

Depends R (>= 3.4.0)

Imports Rcpp (>= 1.0.0)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 6.1.1

Suggests knitr

VignetteBuilder knitr

LazyData true

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-01-23 09:40:03 UTC

R topics documented:

BH_SBVAR *Structural Bayesian Vector Autoregression*

Description

Runs a Structural Bayesian Vector Autoregression model with the method developed by Baumeister and Hamilton (2015, 2017, and 2018).

Usage

```
BH_SBVAR(y, nlags, pA, pdetA = NULL, pH = NULL, pP = NULL,
  pP_sig = NULL, pR_sig = NULL, kappa1 = NULL, itr = 5000,
  burn = 0, thin = 1, acc_irf = TRUE, h1_irf = 12, ci = 0.975)
```

Arguments

y	*(T x n)* matrix containing the endogenous variables. *T* is the number of observations and *n* is the number of endogenous variables.
nlags	Integer specifying the lag order.
pA	*(n x n x 8)* array where *n* is the number of endogenous variables and each slice of the third dimension contains the prior distributions (NA - no prior, 0 - symmetric t-distribution, 1 - non-central t-distribution), sign restrictions (NA - no restriction, 1 - positive restriction, -1 - negative restriction), distribution positions, distribution scales, distribution degrees of freedom, distribution skew, long-run restriction scale parameters, and random-walk proposal scale parameters for the coefficient matrix *A*, respectively.
pdetA	*(1 x 1 x 6)* array where each slice of the third dimension contains the prior distributions (NA - no prior, 0 - symmetric t-distribution, 1 - non-central t-distribution), sign restrictions (NA - no restriction, 1 - positive restriction, -1 - negative restriction), distribution positions, distribution scales, distribution degrees of freedom, and distribution skew parameters for the determinant of *A*, respectively (default = NULL). NULL indicates no priors for the determinant of *A*.
pH	*(n x n x 6)* array where *n* is the number of endogenous variables and each slice of the third dimension contains the prior distributions (NA - no prior, 0 - symmetric t-distribution, 1 - non-central t-distribution), sign restrictions (NA - no restriction, 1 - positive restriction, -1 - negative restriction), distribution positions, distribution scales, distribution degrees of freedom, distribution skew parameters for *H*, the inverse of *A*, respectively (default = NULL). NULL indicates no priors for the inverse of *A*.
pP	*(k x n)* matrix containing the prior position parameters for the reduced form lagged coefficient matrix Φ (default = NULL). $k = nL + 1$, *n* is the number of endogenous variables, and *L* is the lag length. NULL indicates no priors for Φ.

pP_sig	*(k x k)* matrix containing values indicating confidence in the priors for Φ (default = NULL). $k = nL + 1$, *n* is the number of endogenous variables, and *L* is the lag length. NULL indicates no priors for Φ.
pR_sig	*(k x k x n)* array containing values indicating confidence in long-run restrictions on the reduced form lagged coefficient matrix Φ (default = NULL). $k = nL+1$, *n* is the number of endogenous variables, and *L* is the lag length. NULL indicates no long-run restrictions.
kappa1	*(1 x n)* matrix containing values indicating confidence in priors for the structural variances (default = NULL). *n* is the number of endogenous variables. NULL indicates no priors for structural variances.
itr	Integer specifying the total number of iterations for the algorithm (default = 5000).
burn	Integer specifying the number of draws to throw out at the beginning of the algorithm (default = 0).
thin	Integer specifying the thinning parameter (default = 1). All draws beyond burn are kept when thin = 1. Draw 1, draw 3, etc. beyond burn are kept when thin = 2.
acc_irf	Boolean indicating whether accumulated impulse responses are to be returned (default = TRUE).
h1_irf	Integer specifying the time horizon for computing impulse responses (default = 12).
ci	Numeric value indicating credibility intervals for the estimates to be returned (default = 0.975).

Details

Runs a Structural Bayesian Vector Autoregression model with the method developed in Baumeister and Hamilton (2015, 2017, and 2018). The function returns a list containing the results.

Value

A list containing the following:

accept_rate: Acceptance rate of the algorithm.

y and x: Matrices containing the endogenous variables and their lags.

pA, pdetA, pH, pP, pP_sig, pR, pR_sig, tau1, and kappa1: Matrices and arrays containing prior information.

A_start: Matrix containing estimates of the parameters in *A* from the optimization routine.

A, detA, H, B, and Phi: Arrays containing estimates of the model parameters. The first, second, and third slices of the third dimension are lower, median, and upper bounds of the estimates.

HD and IRF: Arrays containing historical decomposition of structural shocks and impulse response functions. The first, second, and third slices of the third dimension are lower, median, and upper bounds of the estimates.

A_den, detA_den, and H_den: Lists containing the horizontal and vertical axis coordinates of posterior densities of *A*, *det(A)*, and *H*.

Line and ACF plots of the estimates for *A*, *det(A)*, and *H*.

Author(s)

Paul Richardson

References

Baumeister, C., and Hamilton, J.D. (2015). Sign restrictions, structural vector autoregressions, and useful prior information. *Econometrica*, 83(5), 1963-1999.

Baumeister, C., and Hamilton, J.D. (2017). Structural interpretation of vector autoregressions with incomplete identification: Revisiting the role of oil supply and demand shocks (No. w24167). National Bureau of Economic Research.

Baumeister, C., and Hamilton, J.D. (2018). Inference in structural vector autoregressions when the identifying assumptions are not fully believed: Re-evaluating the role of monetary policy in economic fluctuations. *Journal of Monetary Economics*,

Examples

```
# Import data
library(BHSBVAR)
set.seed(123)
data(USLMData)
y <- matrix(data = c(USLMData$Wage, USLMData$Employment), ncol = 2)
colnames(y) <- c("Wage", "Employment")

# Set function arguments
nlags <- 4
itr <- 5000
burn <- 0
thin <- 1
acc_irf <- TRUE
h1_irf <- 20
ci <- 0.975

# Priors for A
pA <- array(data = NA, dim = c(2, 2, 8))
pA[, , 1] <- c(0, NA, 0, NA)
pA[, , 2] <- c(1, NA, -1, NA)
pA[, , 3] <- c(0.6, 1, -0.6, 1)
pA[, , 4] <- c(0.6, NA, 0.6, NA)
pA[, , 5] <- c(3, NA, 3, NA)
pA[, , 6] <- c(NA, NA, NA, NA)
pA[, , 7] <- c(NA, NA, 1, NA)
pA[, , 8] <- c(2.4, NA, 2.4, NA)

# Position priors for Phi
pP <- matrix(data = 0, nrow = ((nlags * ncol(pA)) + 1), ncol = ncol(pA))
pP[1:nrow(pA), 1:ncol(pA)] <-
  diag(x = 1, nrow = nrow(pA), ncol = ncol(pA))

# Confidence in the priors for Phi
x1 <-
  matrix(data = NA, nrow = (nrow(y) - nlags),
```

```
         ncol = (ncol(y) * nlags))
for (k in 1:nlags) {
  x1[, (ncol(y) * (k - 1) + 1):(ncol(y) * k)] <-
    y[(nlags - k + 1):(nrow(y) - k),]
}
x1 <- cbind(x1, 1)
colnames(x1) <-
  c(paste(rep(colnames(y), nlags),
          ".L",
          sort(rep(seq(from = 1, to = nlags, by = 1), times = ncol(y)),
               decreasing = FALSE),
          sep = ""),
    "cons")
y1 <- y[(nlags + 1):nrow(y),]
ee <- matrix(data = NA, nrow = nrow(y1), ncol = ncol(y1))
for (i in 1:ncol(y1)) {
  xx <- cbind(x1[, seq(from = i, to = (ncol(x1) - 1), by = ncol(y1))], 1)
  yy <- matrix(data = y1[, i], ncol = 1)
  phi <- solve(t(xx) %*% xx, t(xx) %*% yy)
  ee[, i] <- yy - (xx %*% phi)
}
somega <- (t(ee) %*% ee) / nrow(ee)
lambda0 <- 0.2
lambda1 <- 1
lambda3 <- 100
v1 <- matrix(data = (1:nlags), nrow = nlags, ncol = 1)
v1 <- v1^((-2) * lambda1)
v2 <- matrix(data = diag(solve(diag(diag(somega)))), ncol = 1)
v3 <- kronecker(v1, v2)
v3 <- (lambda0^2) * rbind(v3, (lambda3^2))
v3 <- 1 / v3
pP_sig <- diag(x = 1, nrow = nrow(v3), ncol = nrow(v3))
diag(pP_sig) <- v3

# Confidence in long-run restriction priors
pR_sig <-
  array(data = 0,
        dim = c(((nlags * ncol(y)) + 1),
                ((nlags * ncol(y)) + 1),
                ncol(y)))
Ri <-
  cbind(kronecker(matrix(data = 1, nrow = 1, ncol = nlags),
                  matrix(data = c(1, 0), nrow = 1)),
        0)
pR_sig[, , 2] <- (t(Ri) %*% Ri) / 0.1

# Confidence in priors for D
kappa1 <- matrix(data = 2, nrow = 1, ncol = ncol(y))

# Set graphical parameters
par(cex.axis = 0.8, cex.main = 1, font.main = 1, family = "serif",
    mfrow = c(2, 2), mar = c(2, 2.2, 2, 1), las = 1)
```

```
# Run the model and estimate the model parameters
results1 <-
  BH_SBVAR(y = y, nlags = nlags, pA = pA, pP = pP, pP_sig = pP_sig,
           pR_sig = pR_sig, kappa1 = kappa1, itr = itr, burn = burn,
           thin = thin, acc_irf = acc_irf,
           h1_irf = h1_irf, ci = ci)
```

`Dist_Plots` *Plot Posterior Distributions Against Priors*

Description

Plot Posterior Distributions Against Priors.

Usage

```
Dist_Plots(results, A_titles, H_titles = NULL, xlab = NULL,
  ylab = NULL)
```

Arguments

`results`	List containing the results from running BH_SBVAR().
`A_titles`	*(n x n)* matrix containing the titles for the plots of the estimated parameters in the coefficient matrix *A*. *n* is the number of endogenous variables.
`H_titles`	*(n x n)* matrix containing the titles for the plots of the estimated parameters in the coefficient matrix *H* (default = NULL). *n* is the number of endogenous variables.
`xlab`	Character label for the horizontal axis of historical decomposition plots (default = NULL). Default produces plots without a label for the horizontal axis.
`ylab`	Character label for the vertical axis of historical decomposition plots (default = NULL). Default produces plots without a label for the vertical axis.

Details

Plots posterior distributions against prior distributions.

Author(s)

Paul Richardson

Examples

```
# Import data
library(BHSBVAR)
set.seed(123)
data(USLMData)
y <- matrix(data = c(USLMData$Wage, USLMData$Employment), ncol = 2)
colnames(y) <- c("Wage", "Employment")
```

```
# Set function arguments
nlags <- 4
itr <- 5000
burn <- 0
thin <- 1
acc_irf <- TRUE
h1_irf <- 20
ci <- 0.975

# Priors for A
pA <- array(data = NA, dim = c(2, 2, 8))
pA[, , 1] <- c(0, NA, 0, NA)
pA[, , 2] <- c(1, NA, -1, NA)
pA[, , 3] <- c(0.6, 1, -0.6, 1)
pA[, , 4] <- c(0.6, NA, 0.6, NA)
pA[, , 5] <- c(3, NA, 3, NA)
pA[, , 6] <- c(NA, NA, NA, NA)
pA[, , 7] <- c(NA, NA, 1, NA)
pA[, , 8] <- c(2.4, NA, 2.4, NA)

# Position priors for Phi
pP <- matrix(data = 0, nrow = ((nlags * ncol(pA)) + 1), ncol = ncol(pA))
pP[1:nrow(pA), 1:ncol(pA)] <-
  diag(x = 1, nrow = nrow(pA), ncol = ncol(pA))

# Confidence in the priors for Phi
x1 <-
  matrix(data = NA, nrow = (nrow(y) - nlags),
         ncol = (ncol(y) * nlags))
for (k in 1:nlags) {
  x1[, (ncol(y) * (k - 1) + 1):(ncol(y) * k)] <-
    y[(nlags - k + 1):(nrow(y) - k),]
}
x1 <- cbind(x1, 1)
colnames(x1) <-
  c(paste(rep(colnames(y), nlags),
          ".L",
          sort(rep(seq(from = 1, to = nlags, by = 1), times = ncol(y)),
               decreasing = FALSE),
          sep = ""),
    "cons")
y1 <- y[(nlags + 1):nrow(y),]
ee <- matrix(data = NA, nrow = nrow(y1), ncol = ncol(y1))
for (i in 1:ncol(y1)) {
  xx <- cbind(x1[, seq(from = i, to = (ncol(x1) - 1), by = ncol(y1))], 1)
  yy <- matrix(data = y1[, i], ncol = 1)
  phi <- solve(t(xx) %*% xx, t(xx) %*% yy)
  ee[, i] <- yy - (xx %*% phi)
}
somega <- (t(ee) %*% ee) / nrow(ee)
lambda0 <- 0.2
lambda1 <- 1
lambda3 <- 100
```

```
v1 <- matrix(data = (1:nlags), nrow = nlags, ncol = 1)
v1 <- v1^((-2) * lambda1)
v2 <- matrix(data = diag(solve(diag(diag(somega)))), ncol = 1)
v3 <- kronecker(v1, v2)
v3 <- (lambda0^2) * rbind(v3, (lambda3^2))
v3 <- 1 / v3
pP_sig <- diag(x = 1, nrow = nrow(v3), ncol = nrow(v3))
diag(pP_sig) <- v3

# Confidence in long-run restriction priors
pR_sig <-
  array(data = 0,
        dim = c(((nlags * ncol(y)) + 1),
                ((nlags * ncol(y)) + 1),
                ncol(y)))
Ri <-
  cbind(kronecker(matrix(data = 1, nrow = 1, ncol = nlags),
                  matrix(data = c(1, 0), nrow = 1)),
        0)
pR_sig[, , 2] <- (t(Ri) %*% Ri) / 0.1

# Confidence in priors for D
kappa1 <- matrix(data = 2, nrow = 1, ncol = ncol(y))

# Set graphical parameters
par(cex.axis = 0.8, cex.main = 1, font.main = 1, family = "serif",
    mfrow = c(2, 2), mar = c(2, 2.2, 2, 1), las = 1)

# Run the model and estimate the model parameters
results1 <-
  BH_SBVAR(y = y, nlags = nlags, pA = pA, pP = pP, pP_sig = pP_sig,
           pR_sig = pR_sig, kappa1 = kappa1, itr = itr, burn = burn,
           thin = thin, acc_irf = acc_irf,
           h1_irf = h1_irf, ci = ci)

# Plot Posterior and Prior Densities
A_titles <-
  matrix(data = NA_character_, nrow = dim(pA)[1], ncol = dim(pA)[2])
A_titles[1, 1] <- "Wage Elasticity of Labor Demand"
A_titles[1, 2] <- "Wage Elasticity of Labor Supply"
par(mfcol = c(1, 2))
dist_results <-
  Dist_Plots(results = results1, A_titles = A_titles)
```

HD_Plots *Plot Historical Decompositions*

Description

Plot Historical Decompositions.

Usage

```
HD_Plots(results, varnames, shocknames = NULL, xlab = NULL,
  ylab = NULL, freq, start_date)
```

Arguments

`results`	List containing the results from running BH_SBVAR().
`varnames`	Character vector containing the names of the endogenous variables.
`shocknames`	Character vector containing the names of the shocks.
`xlab`	Character label for the horizontal axis of historical decomposition plots (default = NULL). Default produces plots without a label for the horizontal axis.
`ylab`	Character label for the vertical axis of historical decomposition plots (default = NULL). Default produces plots without a label for the vertical axis.
`freq`	Numeric value indicating the frequency of the data.
`start_date`	Numeric vector indicating the date of the first observation of the endogenous variables included in the model.

Details

Plots historical decompositions and returns a list containing the actual processed data used to create the plots.

Value

A list containing historical decompositions:

Author(s)

Paul Richardson

Examples

```
# Import data
library(BHSBVAR)
set.seed(123)
data(USLMData)
y <- matrix(data = c(USLMData$Wage, USLMData$Employment), ncol = 2)
colnames(y) <- c("Wage", "Employment")

# Set function arguments
nlags <- 4
itr <- 5000
burn <- 0
thin <- 1
acc_irf <- TRUE
h1_irf <- 20
ci <- 0.975

# Priors for A
```

```
pA <- array(data = NA, dim = c(2, 2, 8))
pA[, , 1] <- c(0, NA, 0, NA)
pA[, , 2] <- c(1, NA, -1, NA)
pA[, , 3] <- c(0.6, 1, -0.6, 1)
pA[, , 4] <- c(0.6, NA, 0.6, NA)
pA[, , 5] <- c(3, NA, 3, NA)
pA[, , 6] <- c(NA, NA, NA, NA)
pA[, , 7] <- c(NA, NA, 1, NA)
pA[, , 8] <- c(2.4, NA, 2.4, NA)

# Position priors for Phi
pP <- matrix(data = 0, nrow = ((nlags * ncol(pA)) + 1), ncol = ncol(pA))
pP[1:nrow(pA), 1:ncol(pA)] <-
  diag(x = 1, nrow = nrow(pA), ncol = ncol(pA))

# Confidence in the priors for Phi
x1 <-
  matrix(data = NA, nrow = (nrow(y) - nlags),
         ncol = (ncol(y) * nlags))
for (k in 1:nlags) {
  x1[, (ncol(y) * (k - 1) + 1):(ncol(y) * k)] <-
    y[(nlags - k + 1):(nrow(y) - k),]
}
x1 <- cbind(x1, 1)
colnames(x1) <-
  c(paste(rep(colnames(y), nlags),
          ".L",
          sort(rep(seq(from = 1, to = nlags, by = 1), times = ncol(y)),
               decreasing = FALSE),
          sep = ""),
    "cons")
y1 <- y[(nlags + 1):nrow(y),]
ee <- matrix(data = NA, nrow = nrow(y1), ncol = ncol(y1))
for (i in 1:ncol(y1)) {
  xx <- cbind(x1[, seq(from = i, to = (ncol(x1) - 1), by = ncol(y1))], 1)
  yy <- matrix(data = y1[, i], ncol = 1)
  phi <- solve(t(xx) %*% xx, t(xx) %*% yy)
  ee[, i] <- yy - (xx %*% phi)
}
somega <- (t(ee) %*% ee) / nrow(ee)
lambda0 <- 0.2
lambda1 <- 1
lambda3 <- 100
v1 <- matrix(data = (1:nlags), nrow = nlags, ncol = 1)
v1 <- v1^((-2) * lambda1)
v2 <- matrix(data = diag(solve(diag(diag(somega)))), ncol = 1)
v3 <- kronecker(v1, v2)
v3 <- (lambda0^2) * rbind(v3, (lambda3^2))
v3 <- 1 / v3
pP_sig <- diag(x = 1, nrow = nrow(v3), ncol = nrow(v3))
diag(pP_sig) <- v3

# Confidence in long-run restriction priors
```

```
pR_sig <-
  array(data = 0,
        dim = c(((nlags * ncol(y)) + 1),
                ((nlags * ncol(y)) + 1),
                ncol(y)))
Ri <-
  cbind(kronecker(matrix(data = 1, nrow = 1, ncol = nlags),
                  matrix(data = c(1, 0), nrow = 1)),
        0)
pR_sig[, , 2] <- (t(Ri) %*% Ri) / 0.1

# Confidence in priors for D
kappa1 <- matrix(data = 2, nrow = 1, ncol = ncol(y))

# Set graphical parameters
par(cex.axis = 0.8, cex.main = 1, font.main = 1, family = "serif",
    mfrow = c(2, 2), mar = c(2, 2.2, 2, 1), las = 1)

# Run the model and estimate the model parameters
results1 <-
  BH_SBVAR(y = y, nlags = nlags, pA = pA, pP = pP, pP_sig = pP_sig,
           pR_sig = pR_sig, kappa1 = kappa1, itr = itr, burn = burn,
           thin = thin, acc_irf = acc_irf,
           h1_irf = h1_irf, ci = ci)

# Plot historical decompositions
varnames <- colnames(USLMData)[2:3]
shocknames <- c("Labor Demand","Labor Supply")
freq <- 4
start_date <-
  c(floor(USLMData[(nlags + 1), 1]),
    round(((USLMData[(nlags + 1), 1] %% 1) * freq), digits = 0))
hd_results <-
  HD_Plots(results  = results1, varnames = varnames,
           shocknames = shocknames,
           freq = freq, start_date = start_date)
```

IRF_Plots *Plot Impulse Responses*

Description

Plot Impulse Responses.

Usage

```
IRF_Plots(results, varnames, shocknames = NULL, xlab = NULL,
  ylab = NULL)
```

Arguments

results	List containing the results from running BH_SBVAR().
varnames	Character vector containing the names of the endogenous variables.
shocknames	Character vector containing the names of the shocks.
xlab	Character label for the horizontal axis of impulse response plots (default = NULL). Default produces plots without a label for the horizontal axis.
ylab	Character label for the vertical axis of impulse response plots (default = NULL). Default produces plots without a label for the vertical axis.

Details

Plots impulse responses and returns a list containing the actual processed data used to create the plots.

Value

A list containing impulse responses:

Author(s)

Paul Richardson

Examples

```
# Import data
library(BHSBVAR)
set.seed(123)
data(USLMData)
y <- matrix(data = c(USLMData$Wage, USLMData$Employment), ncol = 2)
colnames(y) <- c("Wage", "Employment")

# Set function arguments
nlags <- 4
itr <- 5000
burn <- 0
thin <- 1
acc_irf <- TRUE
h1_irf <- 20
ci <- 0.975

# Priors for A
pA <- array(data = NA, dim = c(2, 2, 8))
pA[, , 1] <- c(0, NA, 0, NA)
pA[, , 2] <- c(1, NA, -1, NA)
pA[, , 3] <- c(0.6, 1, -0.6, 1)
pA[, , 4] <- c(0.6, NA, 0.6, NA)
pA[, , 5] <- c(3, NA, 3, NA)
pA[, , 6] <- c(NA, NA, NA, NA)
pA[, , 7] <- c(NA, NA, 1, NA)
pA[, , 8] <- c(2.4, NA, 2.4, NA)
```

```
# Position priors for Phi
pP <- matrix(data = 0, nrow = ((nlags * ncol(pA)) + 1), ncol = ncol(pA))
pP[1:nrow(pA), 1:ncol(pA)] <-
  diag(x = 1, nrow = nrow(pA), ncol = ncol(pA))

# Confidence in the priors for Phi
x1 <-
  matrix(data = NA, nrow = (nrow(y) - nlags),
         ncol = (ncol(y) * nlags))
for (k in 1:nlags) {
  x1[, (ncol(y) * (k - 1) + 1):(ncol(y) * k)] <-
    y[(nlags - k + 1):(nrow(y) - k),]
}
x1 <- cbind(x1, 1)
colnames(x1) <-
  c(paste(rep(colnames(y), nlags),
          ".L",
          sort(rep(seq(from = 1, to = nlags, by = 1), times = ncol(y)),
               decreasing = FALSE),
          sep = ""),
    "cons")
y1 <- y[(nlags + 1):nrow(y),]
ee <- matrix(data = NA, nrow = nrow(y1), ncol = ncol(y1))
for (i in 1:ncol(y1)) {
  xx <- cbind(x1[, seq(from = i, to = (ncol(x1) - 1), by = ncol(y1))], 1)
  yy <- matrix(data = y1[, i], ncol = 1)
  phi <- solve(t(xx) %*% xx, t(xx) %*% yy)
  ee[, i] <- yy - (xx %*% phi)
}
somega <- (t(ee) %*% ee) / nrow(ee)
lambda0 <- 0.2
lambda1 <- 1
lambda3 <- 100
v1 <- matrix(data = (1:nlags), nrow = nlags, ncol = 1)
v1 <- v1^((-2) * lambda1)
v2 <- matrix(data = diag(solve(diag(diag(somega)))), ncol = 1)
v3 <- kronecker(v1, v2)
v3 <- (lambda0^2) * rbind(v3, (lambda3^2))
v3 <- 1 / v3
pP_sig <- diag(x = 1, nrow = nrow(v3), ncol = nrow(v3))
diag(pP_sig) <- v3

# Confidence in long-run restriction priors
pR_sig <-
  array(data = 0,
        dim = c(((nlags * ncol(y)) + 1),
                ((nlags * ncol(y)) + 1),
                ncol(y)))
Ri <-
  cbind(kronecker(matrix(data = 1, nrow = 1, ncol = nlags),
                  matrix(data = c(1, 0), nrow = 1)),
        0)
```

```
pR_sig[, , 2] <- (t(Ri) %*% Ri) / 0.1

# Confidence in priors for D
kappa1 <- matrix(data = 2, nrow = 1, ncol = ncol(y))

# Set graphical parameters
par(cex.axis = 0.8, cex.main = 1, font.main = 1, family = "serif",
    mfrow = c(2, 2), mar = c(2, 2.2, 2, 1), las = 1)

# Run the model and estimate the model parameters
results1 <-
  BH_SBVAR(y = y, nlags = nlags, pA = pA, pP = pP, pP_sig = pP_sig,
           pR_sig = pR_sig, kappa1 = kappa1, itr = itr, burn = burn,
           thin = thin, acc_irf = acc_irf,
           h1_irf = h1_irf, ci = ci)

# Plot impulse responses
varnames <- colnames(USLMData)[2:3]
shocknames <- c("Labor Demand","Labor Supply")
irf_results <-
  IRF_Plots(results = results1, varnames = varnames,
            shocknames = shocknames)
```

USLMData *U.S. Labor Market Data*

Description

Quarterly U.S. labor market time-series data. These data are the data used in Baumeister and Hamilton (2015).

Usage

```
data(USLMData)
```

Format

Data frame object that includes "Date", "Wage", and "Employment" variables. These data are the percent change in U.S. real wage and employment and were created by taking the difference of the natural log of U.S. real wage and employment levels and multiplying by 100.

Source

http://econweb.ucsd.edu/~jhamilton/

References

Baumeister, C., and Hamilton, J.D. (2015). Sign restrictions, structural vector autoregressions, and useful prior information. *Econometrica*, 83(5), 1963-1999.

Examples

```
data(USLMData)
```

Index