

# Package ‘HSAR’

October 18, 2017

**Type** Package

**Title** Hierarchical Spatial Autoregressive Model

**Version** 0.4.2

**Date** 2017-10-16

**Author** Guanpeng Dong, Richard Harris, Angelos Mimis

**Maintainer** Angelos Mimis <mimis@panteion.gr>

**Description** A Hierarchical Spatial Autoregressive Model (HSAR), based on a Bayesian Markov Chain Monte Carlo (MCMC) algorithm ( Dong and Harris (2014) <doi:10.1111/gean.12049> ). The creation of this package was supported by the Economic and Social Research Council (ESRC) through the Applied Quantitative Methods Network: Phase II, grant number ES/K006460/1.

**License** GPL (>= 2)

**Depends** R (>= 2.7.0)

**Imports** spdep, Rcpp

**LinkingTo** Rcpp, RcppArmadillo

**Suggests** maptools, rgdal, rgeos, RColorBrewer, classInt, knitr

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2017-10-18 13:36:07 UTC

## R topics documented:

HSAR-package	2
Beijingdistricts	3
hsar	3
landprice	8
landSPDF	9
sar	10
summary.mcmc_hsar	12
summary.mcmc_hsar_lambda_0	13

summary.mcmc_hsar_rho_0 . . . . .	14
summary.mcmc_sar . . . . .	14

<b>Index</b>	<b>16</b>
--------------	-----------

---

HSAR-package	<i>Hierarchical Spatial Autoregressive Model</i>
--------------	--

---

## Description

Implements a Hierarchical Spatial Simultaneous Autoregressive Model (HSAR) or a multi-scale spatial econometrics model, with inference in a Bayesian setting using Markov chain Monte Carlo (MCMC) simulation. The approach is developed for modelling geographic data with a hierarchical/nested structure, for example, houses nesting into districts or fine-grained spatial units into more aggregated units. The HSAR model brings together the spatial econometrics and multilevel models and thus suitable for a simultaneous capturing the potential spatial dependence (autocorrelations) at each level of the data hierarchy arising from geographical proximity effect and the contextual effect (or group dependence effect) from higher-level units upon lower-level units. The creation of this package was supported by the Economic and Social Research Council (ESRC) through the Applied Quantitative Methods Network: Phase II, grant number ES/K006460/1.

## Details

Package: HSAR  
 Type: Package  
 Version: 0.4.2  
 Date: 2017-10-16  
 License: GPL (>= 2)

## Author(s)

Guanpeng Dong, Richard Harris, Angelos Mimis < <mimis@panteion.gr>>

## References

- Anselin, L. (1988). *Spatial Econometrics: Methods and Models*. Dordrecht: Kluwer Academic Publishers.
- Goldstein, H. (2003). *Multilevel Statistical Methods, 3rd ed.* London: Arnold.
- LeSage, J. P., and R. K. Pace. (2009). *Introduction to Spatial Econometrics*. Boca Raton, FL: CRC Press/Taylor & Francis
- Dong, G. and Harris, R. 2015. Spatial Autoregressive Models for Geographically Hierarchical Data Structures. *Geographical Analysis*, 47:173-191.

---

Beijingdistricts      *Boundaries of districts (SpatialPolygonsDataFrame) in Beijing*

---

### Description

The geographic boundaries of districts (SpatialPolygonsDataFrame) in Beijing. This is only a subset of districts in Beijing where our land parcel samples fall into.

### Usage

```
data(Beijingdistricts)
```

### See Also

[landprice](#), [landSPDF](#)

### Examples

```
data(Beijingdistricts)

library(spdep)
plot(Beijingdistricts,border="light grey")
# extract the area of each district
library(rgeos)
library(classInt)
library(RColorBrewer)

Beijingdistricts$geo.area <- gArea(Beijingdistricts,byid=TRUE) / 1000000
x <- Beijingdistricts$geo.area
breaks <- classIntervals(x,4,"fisher")$brks
groups <- cut(x,breaks,include.lowest=TRUE,labels=FALSE)
palette <- brewer.pal(4, "Blues")
plot(Beijingdistricts,col=palette[groups],border="grey")

# extract the district level spatial weights matrix
nb.list <- poly2nb(Beijingdistricts,queen=FALSE)
mat.list <- nb2mat(nb.list,style="W")
M <- as(mat.list,"dgCMatrix")
```

---

hsar

*Hierarchical SAR model estimation*

---

## Description

The specification of a HSAR model is as follows:

$$y_{i,j} = \rho * \mathbf{W}_i * \mathbf{y} + \mathbf{x}'_{i,j} * \beta + \mathbf{z}'_j * \gamma + \theta_j + \epsilon_{i,j}$$

$$\theta_j = \lambda * \mathbf{M}_j * \theta + \mu_j$$

$$\epsilon_{i,j} \sim N(0, \sigma_e^2), \quad \mu_j \sim N(0, \sigma_u^2)$$

where  $i = 1, 2, \dots, n_j$  and  $j = 1, 2, \dots, J$  are indicators of lower- and higher-level spatial units.  $n_j$  is the number of lower-level units in the  $j$ -th higher level unit and  $\sum_{j=1}^J = \mathbf{N}$ .  $\mathbf{x}'_{i,j}$  and  $\mathbf{z}'_j$  represent vectors of lower- and higher-level independent variables.  $\beta$  and  $\gamma$  are regression coefficients to estimate.  $\theta$ , a  $N \times J$  vector of higher-level random effects, also follows a simultaneous autoregressive process.  $\mathbf{W}$  and  $\mathbf{M}$  are two spatial weights matrices (or neighbourhood connection matrices) at the lower and higher levels, defining how spatial units at each level are connected.  $\rho$  and  $\lambda$  are two spatial autoregressive parameters measuring the strength of the dependencies/correlations at the two spatial scales.

A succinct matrix formulation of the model is,

$$\mathbf{y} = \rho * \mathbf{W} * \mathbf{y} + \mathbf{X} * \beta + \mathbf{Z} * \gamma + \Delta * \theta + \epsilon$$

$$\theta = \lambda * \mathbf{M} * \theta + \mu$$

It is also useful to note that the HSAR model nests a standard (random intercept) multilevel model when  $\rho$  and  $\lambda$  are both equal to zero and a standard spatial econometric model when  $\lambda$  and  $\sigma_u^2$  are both equal to zero.

## Usage

```
hsar( formula, data = NULL, W=NULL, M=NULL, Delta,
      burnin=5000, Nsim=10000, thinning = 1, parameters.start = NULL)
```

## Arguments

formula	A symbolic description of the model to fit. A formula for the covariate part of the model using the syntax of the <code>lm()</code> function fitting standard linear regression models. Neither the response variable nor the explanatory variables are allowed to contain NA values.
data	A data frame containing variables used in the formula object.
W	The N by N lower-level spatial weights matrix or neighbourhood matrix where N is the total number of lower-level spatial units. The formulation of W could be based on geographical distances separating units or based on geographical contiguity. To ensure the maximum value of the spatial autoregressive parameter $\rho$ less than 1, W should be row-normalised before running the HSAR model. As

in most cases, spatial weights matrix is very sparse, therefore  $W$  here should be converted to a sparse matrix before imported into the `hsar()` function to save computational burden and reduce computing time. More specifically,  $W$  should be a column-oriented numeric sparse matrices of a `dgCMatrix` class defined in the **Matrix** package. The conversion between a dense numeric matrix and a sparse numeric matrix is made quite convenient through the `Matrix` library.

<code>M</code>	The $J$ by $J$ higher-level spatial weights matrix or neighbourhood matrix where $J$ is the total number of higher-level spatial units. Similar with $W$ , the formulation of $M$ could be based on geographical distances separating units or based on geographical contiguity. To ensure the maximum value of the spatial autoregressive parameter $\lambda$ less than 1, $M$ is also row-normalised before running the HSAR model. As with $W$ , $M$ should also be a column-oriented numeric sparse matrices.
<code>Delta</code>	The $N$ by $J$ random effect design matrix that links the $J$ by 1 higher-level random effect vector back to the $N$ by 1 response variable under investigation. It is simply how lower-level units are grouped into each high-level units with columns of the matrix being each higher-level units. As with $W$ and $M$ , $\delta$ should also be a column-oriented numeric sparse matrices.
<code>burnin</code>	The number of MCMC samples to discard as the burnin period.
<code>Nsim</code>	The total number of MCMC samples to generate.
<code>thinning</code>	MCMC thinning factor.
<code>parameters.start</code>	A list with names "rho", "lambda", "sigma2e", "sigma2u" and "beta" corresponding to initial values for the model parameters $\rho, \lambda, \sigma_e^2, \sigma_u^2$ and the regression coefficients respectively.

## Details

In order to use the `hsar()` function, users need to specify the two spatial weights matrices  $W$  and  $M$  and the random effect design matrix  $\delta$ . However, it is very easy to extract such spatial weights matrices from spatial data using the package **spdep**. Geographic distance-based or contiguity-based spatial weights matrix for both spatial points data and spatial polygons data are available in the **spdep** package.

Before the extraction of  $W$  and  $M$ , it is better to first sort the data using the higher-level unit identifier. Then, the random effect design matrix can be extracted simply (see the following example) and so are the two spatial weights matrices. Make sure the order of higher-level units in the weights matrix  $M$  is in line with that in the  $\delta$  matrix.

Two simpler versions of the HSAR model can also be fitted using the `hsar()` function. The first is a HSAR model with  $\lambda$  equal to zero, indicating an assumption of independence in the higher-level random effect  $\theta$ . The second is a HSAR with  $\rho$  equal to zero, indicating an independence assumption in the outcome variable conditioning on the higher-level random effect. This model is useful in situations where we are interested in the neighbourhood/contextual effect on individual's outcomes and have good reasons to suspect the effect from geographical contexts upon individuals to be dependent. Meanwhile we have no information on how lower-level units are connected.

**Value**

A list object containing:

Mbetas	A vector of estimated mean values of regression coefficients.
SDbetas	The standard deviations of estimated regression coefficients.
Mrho	The estimated mean of the lower-level spatial autoregressive parameter $\rho$ .
SDrho	The standard deviation of the estimated lower-level spatial autoregressive parameter.
Mlambda	The estimated mean of the higher-level spatial autoregressive parameter $\lambda$ .
SDlambda	The standard deviation of the estimated higher-level spatial autoregressive parameter.
Msigma2e	The estimated mean of the lower-level variance parameter $\sigma_e^2$ .
SDsigma2e	The standard deviation of the estimated lower-level variance parameter $\sigma_e^2$ .
Msigma2u	The estimated mean of the higher-level variance parameter $\sigma_u^2$ .
SDsigma2u	The standard deviation of the estimated higher-level variance parameter $\sigma_u^2$ .
Mus	Mean values of $\theta$
SDus	Standard deviation of $\theta$
DIC	The deviance information criterion (DIC) of the fitted model.
pd	The effective number of parameters of the fitted model.
Log_Likelihood	The log-likelihood of the fitted model.
R_Squared	A pseudo R square model fit indicator.
impact_direct	Summaries of the direct impact of a covariate effect on the outcome variable.
impact_indirect	Summaries of the indirect impact of a covariate effect on the outcome variable.
impact_total	Summaries of the total impact of a covariate effect on the outcome variable.

**Author(s)**

Guanpeng Dong, Richard Harris, Angelos Mimis <mimis@panteion.gr>

**References**

Dong, G. and Harris, R. 2015. Spatial Autoregressive Models for Geographically Hierarchical Data Structures. *Geographical Analysis*, 47:173-191.

LeSage, J. P., and R. K. Pace. (2009). *Introduction to Spatial Econometrics*. Boca Raton, FL: CRC Press/Taylor & Francis.

**See Also**

[sar](#), [landprice](#), [Beijingdistricts](#), [landSPDF](#)

**Examples**

```

library(spdep)

# Running the hsar() function using the Beijing land price data
data(landprice)

# load shapefiles of Beijing districts and land parcels
data(Beijingdistricts)
data(landSPDF)

plot(Beijingdistricts,border="green")
plot(landSPDF,add=TRUE,col="red",pch=16,cex=0.8)

# Define the random effect matrix
model.data <- landprice[order(landprice$district.id),]
head(model.data,50)

# the number of individuals within each neighbourhood
MM <- as.data.frame(table(model.data$district.id))
# the total number of neighbourhood, 100
Utotal <- dim(MM)[1]
Unum <- MM[,2]
Uid <- rep(c(1:Utotal),Unum)

n <- nrow(model.data)
Delta <- matrix(0,nrow=n,ncol=Utotal)
for(i in 1:Utotal) {
  Delta[Uid==i,i] <- 1
}
rm(i)
# Delta[1:50,1:10]
Delta <- as(Delta,"dgCMatrix")

# extract the district level spatial weights matrix using the queen's rule
nb.list <- poly2nb(Beijingdistricts)
mat.list <- nb2mat(nb.list,style="W")
M <- as(mat.list,"dgCMatrix")

# extract the land parcel level spatial weights matrix
nb.25 <- dnearneigh(landSPDF,0,2500)
# to a weights matrix
dist.25 <- nbdists(nb.25,landSPDF)
dist.25 <- lapply(dist.25,function(x) exp(-0.5 * (x / 2500)^2))
mat.25 <- nb2mat(nb.25,glist=dist.25,style="W")
W <- as(mat.25,"dgCMatrix")

## run the hsar() function
res.formula <- lnprice ~ lnarea + lndcbd + dsubway + dpark + dele +
popden + crimerate + as.factor(year)

betas= coef(lm(formula=res.formula,data=landprice))

```

```

pars=list( rho = 0.5,lambda = 0.5, sigma2e = 2.0, sigma2u = 2.0, betas = betas )

## Not run:
res <- hsar(res.formula,data=model.data,W=W,M=M,Delta=Delta,
burnin=5000, Nsim=10000, thinning = 1, parameters.start=pars)
summary(res)

# visualise the district level random effect
library(classInt)
library(RColorBrewer)
x <- as.numeric(res$Mus)
breaks <- classIntervals(x,4,"fisher")$brks
groups <- cut(x,breaks,include.lowest=TRUE,labels=FALSE)
palette <- brewer.pal(4, "Blues")
plot(Beijingdistricts,col=palette[groups],border="grey")

## End(Not run)

```

---

landprice

*Leased residential land parcels, from 2003 to 2009 in Beijing, China*


---

## Description

The residential land parcel data leased to real estate developers from the government during 2003 to 2009. The data contains price information and a range of locational and neighbourhood characteristics for each land parcel. There are 1117 samples in the data after dropping those without price information.

## Usage

```
data(landprice)
```

## Format

A data frame with 1117 observations on the following 11 variables.

**obs** An unique identifier for each land parcel.

**lnprice** The log of the leasing price per square metre of each residential land parcel (unit: RMB, Chinese yuan)

**dsubway** The log of the distance of each land parcel to the nearest railway station (unit:meters)

**dele** The log of the distance of each land parcel to the nearest elementary school (unit:meters)

**dpark** The log of the distance of each land parcel to the nearest green park (unit:meters)

**lnarea** The log of the size of each land parcel (unit: square meters).

**lndcbd** The log of the distance of each land parcel to the CBD (centre business district) in Beijing (unit:meters)



**year** The year when each land parcel was leased with values of 0,1,2,3,4,5,6 representing year 2003,2004,2005,2006,2007,2008,2009

**popden** The population density of each district (unit: 1000 persons per square kilometers)

**crimerate** The number of reported serious crimes committed in each district per 1000 persons.

**district.id** The identifier of the district where each land parcel is located.

## References

Harris, R., G.P. Dong, and W.Z. Zhang. (2013).Using Contextualised Geographically Weighted Regression to model the Spatial Heterogeneity of Land Prices in Beijing, China. *Transaction in GIS 17(6)*, 901-19.

Dong G, Harris R, Jones K, Yu J (2015) Multilevel Modelling with Spatial Interaction Effects with Application to an Emerging Land Market in Beijing, China. *PLoS ONE* 10(6): e0130761. doi:10.1371/journal.pone.0130761

## See Also

[Beijingdistricts](#), [landSPDF](#)

## Examples

```
data(landprice)
head(landprice)
```

---

landSPDF

*Beijing land price data*

---

## Description

The spatial locations (`SpatialPointsDataFrame`) of the Beijing land price data. It is used to extract the spatial weights matrix among land parcels.

## Usage

```
data(landSPDF)
```

## See Also

[landprice](#), [Beijingdistricts](#)

## Examples

```

data(landSPDF)

library(spdep)
plot(landSPDF,col="red",pch=16,cex=0.7)

data(landprice)

library(classInt)
library(RColorBrewer)

# link the variables in the landprice to the spatial data
index.match <- match(landSPDF$obs,landprice$obs)
landSPDF@data <- data.frame(landSPDF@data,landprice[index.match,])

par(mar=c(0,0,0,0))
x <- landSPDF$lnprice
breaks <- classIntervals(x,4,"fisher")$brks
groups <- cut(x,breaks,include.lowest=TRUE,labels=FALSE)
palette <- brewer.pal(4, "Blues")
plot(landSPDF,pch=19,col=palette[groups],cex=0.8)

# extract a spatial weights matrix based on the distances between pairs of land parcels
nb.25 <- dnearneigh(landSPDF,0,2500)
# to a weights matrix
dist.25 <- nbdists(nb.25,landSPDF)
dist.25 <- lapply(dist.25,function(x) exp(-0.5 * (x / 2500)^2))
mat.25 <- nb2mat(nb.25,glis=dist.25,style="W")
W <- as(mat.25,"dgCMatrix")

```

---

sar

*SAR model estimation*

---

## Description

The `sar()` function implements a standard spatial econometrics model (SAR) or a spatially lagged dependent variable model using the Markov chain Monte Carlo (MCMC) simulation approach.

## Usage

```

sar( formula, data = NULL, W, burnin=5000, Nsim=10000,
      thinning=1, parameters.start = NULL )

```

## Arguments

formula	A symbolic description of the model to fit. A formula for the covariate part of the model using the syntax of the <code>lm()</code> function fitting standard linear regression models. Neither the response variable nor the explanatory variables are allowed to contain NA values.
---------	---

<code>data</code>	A data frame containing variables used in the formula object.
<code>W</code>	The N by N spatial weights matrix or neighbourhood matrix where N is the number of spatial units. The formulation of W could be based on geographical distances separating units or based on geographical contiguity. To ensure the maximum value of the spatial autoregressive parameter $\rho$ less than 1, W is usually row-normalised before implementing the SAR model. As in most cases, spatial weights matrix is very sparse, therefore W here should be converted to a sparse matrix before imported into the <code>sar()</code> function to save computational burdon and reduce computing time. More specifically, W should be a column-oriented numeric sparse matrices of a <code>dgMatrix</code> class defined in the <b>Matrix</b> package. The converion between a dense numeric matrix and a sparse numeric matrix is made quite convenient through the Matrix library.
<code>burnin</code>	The number of MCMC samples to discard as the burnin period.
<code>Nsim</code>	The total number of MCMC samples to generate.
<code>thinning</code>	MCMC thinning factor.
<code>parameters.start</code>	A list with names "rho", "sigma2e", and "beta" corresponding to initial values for the model parameters $\rho$ , $\sigma_e^2$ and the regression coefficients respectively.

### Value

A list object containing:

<code>Mbetas</code>	A vector of estimated mean values of regression coefficients.
<code>SDbetas</code>	The standard deviations of estimated regression coefficients.
<code>Mrho</code>	The estimated mean of the lower-level spatial autoregressive parameter $\rho$ .
<code>SDrho</code>	The standard deviation of the estimated lower-level spatial autoregressive parameter.
<code>Msigma2e</code>	The estimated mean of the lower-level variance parameter $\sigma_e^2$ .
<code>SDsigma2e</code>	The standard deviation of the estimated lower-level variance parameter $\sigma_e^2$ .
<code>DIC</code>	The deviance information criterion (DIC) of the fitted model.
<code>pd</code>	The effective number of parameters of the fitted model.
<code>Log_Likelihood</code>	The log-likelihood of the fitted model.
<code>R_Squared</code>	A pseudo R square model fit indicator.
<code>impact_direct</code>	Summaries of the direct impact of a covariate effect on the outcome variable.
<code>impact_idirect</code>	Summaries of the indirect impact of a covariate effect on the outcome variable.
<code>impact_total</code>	Summaries of the total impact of a covariate effect on the outcome variable.

### Author(s)

Guanpeng Dong, Richard Harris, Angelos Mimis <mimis@panteion.gr>

**References**

- Anselin, L. (1988). *Spatial Econometrics: Methods and Models*. Dordrecht: Kluwer Academic Publishers.
- LeSage, J. P., and R. K. Pace. (2009). *Introduction to Spatial Econometrics*. Boca Raton, FL: CRC Press/Taylor & Francis

**See Also**

[hsar](#), [landprice](#), [Beijingdistricts](#), [landSPDF](#)

**Examples**

```
data(landprice)
head(landprice)

data(landSPDF)

# extract the land parcel level spatial weights matrix
library(spdep)
nb.25 <- dnearneigh(landSPDF,0,2500)
# to a weights matrix
dist.25 <- nbdists(nb.25,landSPDF)
dist.25 <- lapply(dist.25,function(x) exp(-0.5 * (x / 2500)^2))
mat.25 <- nb2mat(nb.25,glist=dist.25,style="W")
W <- as(mat.25,"dgCMatrix")

## run the sar() function
res.formula <- lnprice ~ lnarea + lndcbd + dsubway + dpark + dele +
popden + crimerate + as.factor(year)

betas= coef(lm(formula=res.formula,data=landprice))

pars=list( rho = 0.5, sigma2e = 2.0, betas = betas )

## Not run:
res <- sar(res.formula,data=landprice,W=W,
burnin=5000, Nsim=10000, thinning=1, parameters.start=pars)
summary(res)

## End(Not run)
```

---

summary.mcmc\_hsar

*summary method for class mcmc\_hsar*


---

**Description**

Methods for presenting the result of a fitted HSAR model.

**Usage**

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

**Arguments**

object	mcmc_hsar	An mcmc_hsar object returned from the hsar function
x	mcmc_hsar	An mcmc_hsar object returned from the hsar function
...		Other arguments passed through

**See Also**

[hsar](#)

---

summary.mcmc\_hsar\_lambda\_0

*summary method for class mcmc\_hsar\_lambda\_0*

---

**Description**

Methods for presenting the result of a fitted HSAR model with  $\lambda$  equal to 0. This is a model without the consideration of the possible interdependency between higher-level spatial units.

**Usage**

```
## S3 method for class 'mcmc_hsar_lambda_0'
summary(object, ...)
## S3 method for class 'mcmc_hsar_lambda_0'
print(x, ...)
```

**Arguments**

object	mcmc_hsar_lambda_0	An mcmc_hsar_lambda_0 object returned from the hsar function
x	mcmc_hsar_lambda_0	An mcmc_hsar_lambda_0 object returned from the hsar function
...		Other arguments passed through

**See Also**

[hsar](#)

---

```
summary.mcmc_hsar_rho_0
```

*summary method for class mcmc\_hsar\_rho\_0*

---

### Description

Methods for presenting the result of a fitted HSAR model with  $\rho$  equal to 0. This is a model without the consideration of the possible interdependency between lower-level spatial units.

### Usage

```
## S3 method for class 'mcmc_hsar_rho_0'
summary(object, ...)
## S3 method for class 'mcmc_hsar_rho_0'
print(x, ...)
```

### Arguments

object	mcmc_hsar_rho_0 An mcmc_hsar_rho_0 object returned from the hsar function
x	mcmc_hsar_rho_0 An mcmc_hsar_rho_0 object returned from the hsar function
...	Other arguments passed through

### See Also

[hsar](#)

---

```
summary.mcmc_sar
```

*summary method for class mcmc\_sar*

---

### Description

Methods for presenting the result of a fitted standard spatial econometric model or a spatially lagged dependent variable model

### Usage

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

**Arguments**

object	mcmc_sar	An mcmc_sar object returned from the sar function
x	mcmc_sar	An mcmc_sar object returned from the sar function
...		Other arguments passed through

**See Also**

[sar](#)

# Index

## \*Topic **hierarchical spatial data**

Beijingdistricts, [3](#)  
landprice, [8](#)  
landSPDF, [9](#)

## \*Topic **spatial econometrics**

hsar, [3](#)  
HSAR-package, [2](#)  
sar, [10](#)  
summary.mcmc\_hsar, [12](#)  
summary.mcmc\_hsar\_lambda\_0, [13](#)  
summary.mcmc\_hsar\_rho\_0, [14](#)  
summary.mcmc\_sar, [14](#)

Beijingdistricts, [3](#), [6](#), [9](#), [12](#)

HSAR (HSAR-package), [2](#)

hsar, [3](#), [12–14](#)

HSAR-package, [2](#)

landprice, [3](#), [6](#), [8](#), [9](#), [12](#)

landSPDF, [3](#), [6](#), [9](#), [9](#), [12](#)

print.mcmc\_hsar (summary.mcmc\_hsar), [12](#)

print.mcmc\_hsar\_lambda\_0  
(summary.mcmc\_hsar\_lambda\_0),  
[13](#)

print.mcmc\_hsar\_rho\_0  
(summary.mcmc\_hsar\_rho\_0), [14](#)

print.mcmc\_sar (summary.mcmc\_sar), [14](#)

sar, [6](#), [10](#), [15](#)

summary.mcmc\_hsar, [12](#)

summary.mcmc\_hsar\_lambda\_0, [13](#)

summary.mcmc\_hsar\_rho\_0, [14](#)

summary.mcmc\_sar, [14](#)