Package: krige (via r-universe)

October 27, 2026

Type Package

Title Geospatial Kriging with Metropolis Sampling

Version 0.6.2

Date 2022-04-28

Author Jason S. Byers [aut, cre], Le Bao [aut], Jamie Carson [aut], Jeff Gill [aut]

Maintainer Jason S. Byers <jaybyers55@gmail.com>

Description Estimates kriging models for geographical point-referenced data. Method is described in Gill (2020) <doi:10.1177/1532440020930197>.

License GPL (>= 2)

Depends R (>= 4.0)

Imports Rcpp, stats, graphics, coda

LinkingTo Rcpp

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation yes

Repository https://cran.r-universe.dev

Date/Publication 2022-05-01 13:20:02 UTC

RemoteUrl https://github.com/cran/krige

RemoteRef HEAD

RemoteSha f04f1e628cf17f6133c0f1783253bc48d148f589

Contents

as.matrix.semivariance 2
as.mcmc.krige 3
burnin 5

congCombined 5
ContrivedData 7
exponential.semivariance 8
geweke 10
heidel.welch 11
krige 13
krige.posterior 13
lowerCombined 15
mcmc.samples 17
metropolis.krige 18
NY_subset 22
NYcity_subset 25
predict.krige 27
semivariance 29
semivariogram 32
stateCombined 35
summary.krige 36
update.krige 37
upperCombined 39
WVwells 40

Index **44**

`as.matrix.semivariance` *Convert semivariance to a matrix object*

Description

Convert semivariance to a matrix object

Usage

```
## S3 method for class 'semivariance'
as.matrix(x, ...)

## S3 method for class 'semivariance'
as.data.frame(x, ...)
```

Arguments

`x`	An `semivariance` object created by `semivariance` or `exponential.semivariance`.
`...`	Additional arguments passed to `as.matrix` and `as.data.frame` methods. Not supported for `semivariance` object.

Details

The defaults of semivariance methods give a list or numeric vector. These methods can convert the semivariance output list and vector to `matrix` or `data.frame`.

Value

A matrix containing the computed distance and semivariance.

Examples

```
## Not run:
# Summarize Data
summary(ContrivedData)

# Empirical semivariance for variable y
raw.var <- semivariance(x=ContrivedData$y,coords = cbind(ContrivedData$s.1,
  ContrivedData$s.2))
as.matrix(raw.var)

# Estimation using metropolis.krige()
#' # Set seed
set.seed(1241060320)

M <- 100
#M<-10000

contrived.run <- metropolis.krige(y ~ x.1 + x.2, coords = c("s.1","s.2"),
   data = ContrivedData, n.iter = M, range.tol = 0.05)

# Parametric semivariance
parametric.var <- semivariance(contrived.run)
as.matrix(parametric.var)
as.data.frame(parametric.var)

## End(Not run)
```

---

as.mcmc.krige *Convert* krige *object to an* mcmc *object*

Description

Convert MCMC matrix of posterior samples for use with the **coda** package

Usage

```
## S3 method for class 'krige'
as.mcmc(x, start = 1, end = x$n.iter, thin = 1, ...)

## S3 method for class 'summary.krige'
as.mcmc(x, start = 1, end = x$n.iter, thin = 1, ...)
```

Arguments

x	An krige or summary.krige object.
start	The iteration number of the first observation.
end	The iteration number of the last observation.
thin	The thinning interval between consecutive observations.
...	Additional arguments to be passed to mcmc() methods of coda package.

Details

The function converts a krige output object to a Markov Chain Monte Carlo (mcmc) object used in coda as well as a variety of MCMC packages. It extracts the MCMC matrix of posterior samples from the output of metropolis.krige for further use with other MCMC packages and functions.

Value

A mcmc object.

See Also

coda::as.mcmc()

Examples

```
## Not run:
# Summarize Data
summary(ContrivedData)

# Set seed
set.seed(1241060320)

#For simple illustration, we set to few iterations.
#In this case, a 10,000-iteration run converges to the true parameters.
#If you have considerable time and hardware, delete the # on the next line.
#10,000 iterations took 39 min. with 8 GB RAM & a 1.5 GHz Quad-Core processor.
M <- 100
#M<-10000

contrived.run <- metropolis.krige(y ~ x.1 + x.2, coords = c("s.1","s.2"),
                                  data = ContrivedData, n.iter = M, n.burnin = 20,
                                  range.tol = 0.05)

# Convert to mcmc object
mcmc.contrived.run <- as.mcmc(contrived.run)
#mcmc.contrived.run <- as.mcmc(summary(contrived.run))

# Diagnostics using MCMC packages
coda::raftery.diag(mcmc.contrived.run)
# superdiag::superdiag(mcmc.contrived.run) #NOT WORKING YET

## End(Not run)
```

burnin *Discard Burn-in Period of Kriging Model*

Description

Discard burn-in period of a estimated model from `metropolis.krige`

Usage

```
burnin(object, n.burnin)

## S3 method for class 'krige'
burnin(object, n.burnin = object$n.iter/2)

## S3 method for class 'matrix'
burnin(object, n.burnin = nrow(object)/2)
```

Arguments

`object`	An `krige` object from the `metropolis.krige` function.
`n.burnin`	The number of burnin iterations. Defaults to half of the iterations.

Details

The function discard the burn-in period from the results of `metropolis.krige`. It is generally used for discarding burn-in for `krige` object that keeps all the iterations.

congCombined *Congressional District Public Opinion Ideology in 2010*

Description

These data present measures of ideology in 2010 for 434 districts for the U.S. House of Representatives, recorded as the variable `krige.cong`. Forecasts are based on a kriging model fitted over the 2008 Cooperative Congressional Election Survey (CCES), paired with predictive data from the 2010 Census. Each district's public ideology is paired with the DW-NOMINATE common space score of each of its representative in 2011 (update from McCarty, Poole and Rosenthal 1997). Eight districts have repeated observations in order to include the DW-NOMINATE score when a member was replaced mid-term.

Format

The congCombined dataset has 442 observations and 12 variables. 4 34 out of 435 congressional districts are covered, with eight districts duplicated when a member was replaced mid-term.

stateCD Unique identifier for each congressional district by state. The first two digits are STATEA, and the second two are cd.

krige.cong The ideology of the average citizen in the congressional district.

krige.state.var The variance of ideology among the district's citizens.

cong The term of Congress studied–112 for this dataset.

idno Identification number for the House member–ICPSR numbers continued by Poole & Rosenthal.

state The ICPSR code for the state.

cd The congressional district number.

statenm The first seven letters of the state's name.

party Political party of the House member. 100=Democrat, 200=Republican.

name Last name of the House member, followed by first name if ambiguous.

dwnom1 First dimension DW-NOMINATE common space score for the House member. Higher values are usually interpreted as more right-wing, with lower values as more left-wing.

STATEA The FIPS code for the state.

Source

Ansolabehere, Stephen. 2011. "CCES, Common Content, 2008." Ver. 4.

McCarty, Nolan M., Keith T. Poole and Howard Rosenthal. 1997. *Income Redistribution and the Realignment of American Politics*. American Enterprise Institude Studies on Understanding Economic Inequality. Washington: AEI Press.

Minnesota Population Center. 2011. *National Historical Geographic Information System: Version 2.0.* Minneapolis, MN: University of Minnesota. 'https://www.nhgis.org'

References

Jeff Gill. 2020. Measuring Constituency Ideology Using Bayesian Universal Kriging. *State Politics & Policy Quarterly*. doi:10.1177/1532440020930197

Examples

```
# Descriptive Statistics
summary(congCombined)

# Correlate House Members' DW-NOMINATE Scores with Public Opinion Ideology
cor(congCombined$dwnom1,congCombined$krige.cong)

# Plot House Members' DW-NOMINATE Scores against Public Opinion Ideology
plot(y=congCombined$dwnom1,x=congCombined$krige.cong,
  xlab="District Ideology (Kriging)", ylab="Legislator Ideology (1st Dim., Common Space)",
   main="U.S. House of Representatives", type="n")
```

```
points(y=congCombined$dwnom1[congCombined$party==200],
   x=congCombined$krige.cong[congCombined$party==200],pch="R",col="red")
   points(y=congCombined$dwnom1[congCombined$party==100],
   x=congCombined$krige.cong[congCombined$party==100],pch="D",col="blue")
```

`ContrivedData` *Contrived Example Data*

Description

These data are a simulated point-referenced geospatial data that serve to provide a clean example of a kriging model. There are 500 observations with coordinates located on a unit square.

Format

The `ContrivedData` dataset has 500 observations and 5 variables.

`y` The outcome variable. Its true population functional form is $y_s = 0 + 1x_{1s} + 2x_{2s} + \omega_s + \epsilon_s$. The true variance of ω is $\sigma^2 = 0.5$ and of ϵ is $\tau^2 = 0.5$. The decay term that shapes spatial correlation levels is $\phi = 2.5$.

`x.1` A predictor with a standard uniform distribution.

`x.2` A predictor with a standard normal distribution.

`s.1` Coordinate in eastings for each observation, distributed standard uniform.

`s.2` Coordinate in northings for each observation, distributed standard uniform.

Examples

```
## Not run:
# Summarize example data
summary(ContrivedData)

# Initial OLS model
contrived.ols<-lm(y~x.1+x.2,data=ContrivedData)
# summary(contrived.ols)

# Set seed
set.seed(1241060320)

#For simple illustration, we set to few iterations.
#In this case, a 10,000-iteration run converges to the true parameters.
#If you have considerable time and hardware, delete the # on the next line.
#10,000 iterations took 39 min. with 8 GB RAM & a 1.5 GHz Quad-Core processor.
M <- 100
#M<-10000

contrived.run <- metropolis.krige(y ~ x.1 + x.2, coords = c("s.1","s.2"),
   data = ContrivedData, n.iter = M, n.burnin=20, range.tol = 0.05)
# Alternatively, use burnin() after estimation
```

```
#contrived.run <- burnin(contrived.run, n.burnin=20)

# Summarize the results and examine results against true coefficients
summary(contrived.run)
(TRUTH<-c(0.5,2.5,0.5,0,1,2))

## End(Not run)
```

`exponential.semivariance` *Parametric Exponential Semivariance*

Description

This function returns the value of a parametric powered exponential semivariogram given the values of the parameters and the distance between observations.

Usage

```
exponential.semivariance(...)

## S3 method for class 'krige'
exponential.semivariance(object, ...)

## Default S3 method:
exponential.semivariance(nugget, decay, partial.sill, distance, power = 2, ...)
```

Arguments

`...`	Additional arguments
`object`	A `krige` object of which the values of the estimates are used to calculate the exponential semivariance.
`nugget`	The value of the non-spatial variance, or nugget term.
`decay`	The value of the decay term that sets the level of correlation given distance.
`partial.sill`	The value of the spatial variance, or partial sill term.
`distance`	The distance among observations for which the semivariance value is desired.
`power`	The exponent specified in the powered exponential semivariogram. Defaults to 2, which corresponds to a Gaussian semivariance function.

Details

The models estimated by the `krige` package assume a powered exponential covariance structure. Each parametric covariance function for kriging models corresponds to a related semivariance function, given that highly correlated values will have a small variance in differences while uncorrelated values will vary widely. More specifically, semivariance is equal to half of the variance of the difference in a variable's values at a given distance. That is, the semivariance is defined as:

$\gamma(h) = 0.5 * E[X(s+h) - X(s)]^2$, where X is the variable of interest, s is a location, and h is the distance from s to another location.

The powered exponential covariance structure implies that the semivariance follows the specific functional form of $\gamma(d) = \tau^2 + \sigma^2(1 - \exp(-|\phi d|^p))$ (Banerjee, Carlin, and Gelfand 2015, 27). A perk of this structure is that the special case of *p=1* implies the commonly-used exponential semivariogram, and the special case of *p=2* implies the commonly-used Gaussian semivariogram. Upon estimating a model, it is advisable to graph the functional form of the implied parametric semivariance structure. By substituting estimated values of the `nugget`, `decay`, and `partial.sill` terms, as well as specifying the correct `power` argument, it is possible to compute the implied semivariance from the model. The `distance` argument easily can be a vector of observed distance values.

Value

A semivariance object. It will be a numeric vector with each bin's value of the semivariance.

References

Sudipto Banerjee, Bradley P. Carlin, and Alan E. Gelfand. 2015. *Hierarchical Modeling and Analysis for Spatial Data*. 2nd ed. Boca Raton, FL: CRC Press.

See Also

`semivariogram`, `plot.semivariance`, `exponential.semivariance`

Examples

```
## Not run:
# Summarize data
summary(ContrivedData)

# Set seed
set.seed(1241060320)

M <- 100

contrived.run <- metropolis.krige(y ~ x.1 + x.2, coords = c("s.1","s.2"),
  data = ContrivedData, n.iter = M, range.tol = 0.05)

# Parametric powered exponential semivariogram
exponential.semivariance(contrived.run)

#OLS Model for Residuals
contrived.ols<-lm(y~x.1+x.2,data=ContrivedData)

# Residual semivariance
(resid.semivar <- semivariance(contrived.ols, coords = c("s.1", "s.2"), terms = "residual"))

# Parametric exponential semivariance
exponential.semivariance(nugget=0.5,decay=2.5,partial.sill=0.5,
                         distance=as.numeric(names(resid.semivar)))
```

```
## End(Not run)
```

---

geweke *Geweke Diagnostic for MCMC*

Description

Conducts a Geweke convergence diagnostic on MCMC iterations.

Usage

```
geweke(object, early.prop = 0.1, late.prop = 0.5, precision = 4)

## S3 method for class 'krige'
geweke(object, early.prop = 0.1, late.prop = 0.5, precision = 4)

## S3 method for class 'summary.krige'
geweke(object, early.prop = 0.1, late.prop = 0.5, precision = 4)

## Default S3 method:
geweke(object, early.prop = 0.1, late.prop = 0.5, precision = 4)
```

Arguments

`object`	An matrix or `krige`/`summary.krige` object for which a Geweke diagnostic is desired
`early.prop`	Proportion of iterations to use from the start of the chain.
`late.prop`	Proportion of iterations to use from the end of the chain.
`precision`	Number of digits of test statistics and p-values to print.

Details

This is a generic function currently works with `matrix`, `krige`, and `summary.krige` objects. It is a simplified version of the Geweke test for use with this package.

Geweke's (1992) test for nonconvergence of a MCMC chain is to conduct a difference-of-means test that compares the mean early in the chain to the mean late in the chain. If the means are significantly different from each other, then this is evidence that the chain has not converged. The difference-of-means test is a simple z-ratio, though the standard error is estimated using the spectral density at zero to account for autocorrelation in the chain.

Value

A `matrix` in which the first row consists of z-scores for tests of equal means for the first and last parts of the chain. The second row consists of the corresponding p-values. Each column of the matrix represents another parameter. For each column, a significant result is evidence that the chain has not converged for that parameter. Thus, a non-significant result is desired.

References

John Geweke. 1992. "Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments." In *Bayesian Statistics 4*, ed. J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith. Oxford: Clarendon Press.

See Also

geweke.krige, geweke.summary.krige

Examples

```
## Not run:
# Load Data
data(ContrivedData)

# Set seed
set.seed(1241060320)

M <- 100

contrived.run <- metropolis.krige(y ~ x.1 + x.2, coords = c("s.1","s.2"),
  data = ContrivedData, n.iter = M, range.tol = 0.05)

geweke(contrived.run, early.prop=0.5)
geweke(summary(contrived.run), early.prop=0.5)
geweke(contrived.run$mcmc.mat, early.prop=0.5)
# Note that the default proportions in the geweke() is more typical for longer run.

## End(Not run)
```

---

heidel.welch *Heidelberger and Welch Diagnostic for MCMC*

Description

Conducts a Heidelberger and Welch convergence diagnostic on MCMC iterations.

Usage

```
heidel.welch(object, pvalue)

## S3 method for class 'krige'
heidel.welch(object, pvalue = 0.05)

## S3 method for class 'summary.krige'
heidel.welch(object, pvalue = 0.05)

## Default S3 method:
heidel.welch(object, pvalue = 0.05)
```

Arguments

`object`	An matrix or `krige`/`summary.krige` object for which a Heidelberger and Welch diagnostic is desired
`pvalue`	Alpha level for significance tests. Defaults to 0.05.

Details

This is a generic function currently works with `matrix`, `krige`, and `summary.krige` objects. It is a simplified version of the Heidelberger and Welch test for use with this package.

This is an adaptation of a function in Plummer et al.'s coda package. Heidelberger and Welch's (1993) test for nonconvergence. This version of the diagnostic only reports a Cramer-von Mises test and its corresponding p-value to determine if the chain is weakly stationary with comparisons of early portions of the chain to the end of the chain.

Value

A `matrix` in which the first row consists of the values of the Cramer-von Mises test statistic for each parameter, and the second row consists of the corresponding p-values. Each column of the matrix represents another parameter of interest. A significant result serves as evidence of nonconvergence, so non-significant results are desired.

References

Philip Heidelberger and Peter D. Welch. 1993. "Simulation Run Length Control in the Presence of an Initial Transient." *Operations Research* 31:1109-1144.

Martyn Plummer, Nicky Best, Kate Cowles and Karen Vines. 2006. "CODA: Convergence Diagnosis and Output Analysis for MCMC." *R News* 6:7-11.

See Also

`heidel.welch.krige`, `heidel.welch.summary.krige`, `geweke`

Examples

```
## Not run:
# Load Data
data(ContrivedData)

# Set seed
set.seed(1241060320)

M <- 100

contrived.run <- metropolis.krige(y ~ x.1 + x.2, coords = c("s.1","s.2"),
  data = ContrivedData, n.iter = M, n.burnin = 20, range.tol = 0.05)

heidel.welch(contrived.run)
heidel.welch(summary(contrived.run))
heidel.welch(contrived.run$mcmc.mat)
```

```
## End(Not run)
```

---

krige *krige: Geospatial Kriging with Metropolis Sampling*

Description

Estimates kriging models for geographical point-referenced data. Method is described ' in Gill (2020) <doi:10.1177/1532440020930197>.

---

krige.posterior *Posterior Distribution for the Kriging Process*

Description

This function finds the posterior density of a geospatial linear regression model given a point-referenced geospatial dataset and a set of parameter values. The function is useful for finding the optimum of or for sampling from the posterior distribution.

Usage

```
krige.posterior(
  tau2,
  phi,
  sigma2,
  beta,
  y,
  X,
  east,
  north,
  semivar.exp = 2,
  p.spatial.share = 0.5,
  p.range.share = 0.5,
  p.range.tol = 0.05,
  p.beta.var = 10,
  tot.var = var(y),
  local.Sigma = NULL,
  max.distance = NULL
)
```

Arguments

tau2	Value of the nugget, or non-spatial error variance.
phi	Value of the decay term, driving the level of spatial correlation.
sigma2	Value of the partial sill, or maximum spatial error variance.
beta	Coefficients from linear model.
y	The outcome variable that is used in the kriging model.
X	The matrix of explanatory variables used in the kriging model.
east	Vector of eastings for all observations.
north	Vector of northings for all observations.
semivar.exp	This exponent, which must be greater than 0 and less than or equal to 2, specifies a powered exponential correlation structure for the data. One widely used specification is setting this to 1, which yields an exponential correlation structure. Another common specification is setting this to 2 (the default), which yields a Gaussian correlation structure.
p.spatial.share	Prior for proportion of unexplained variance that is spatial in nature. Must be greater than 0 and less than 1. Defaults to an even split.
p.range.share	Prior for the effective range term, as a proportion of the maximum distance in the data. Users should choose the proportion of distance at which they think the spatial correlation will become negligible. Must be greater than 0. Values greater than 1 are permitted, but users should recognize that this implies that meaningful spatial correlation would persist outside of the convex hull of data. Defaults to half the maximum distance.
p.range.tol	Tolerance term for setting the effective range. At the distance where the spatial correlation drops below this term, it is judged that the effective range has been met. Users are typically advised to leave this at its default value of 0.05 unless they have strong reasons to choose another level. Must be greater than 0 and less than 1.
p.beta.var	Prior for the variance on zero-meaned normal priors on the regression coefficients. Defaults to 10.
tot.var	Combined variance between the nugget and partial sill. Defaults to the variance of y. The metropolis.krige function inserts the residual variance from a standard linear model.
local.Sigma	The user is advised to ignore this option, or leave it the value of NULL. This option is included to reduce the number of computations required when this function is called by metropolis.krige.
max.distance	The user is advised to ignore this option, or leave it the value of NULL. This option is included to reduce the number of computations required when this function is called by metropolis.krige.

Details

This function finds the posterior density for a kriging model. It is designed to be an internal function but is exported in the hope of it can be useful to some users. The function utilizes information

provided about the parameters tau2, phi, sigma2, and beta. It also utilizes the observed data y, X, east, and north. Given a set of parameter values as well as the observed data, the function returns the posterior density for the specified model.

Value

A single number that is the posterior density of the function, which is stored in object of class matrix.

References

Jeff Gill. 2020. Measuring Constituency Ideology Using Bayesian Universal Kriging. *State Politics & Policy Quarterly*. doi:10.1177/1532440020930197

Examples

```
# Summarize Data
summary(ContrivedData)

#Initial OLS Model
contrived.ols<-lm(y~x.1+x.2,data=ContrivedData);summary(contrived.ols)

#Define Covariate Matrix
covariates<-cbind(1,ContrivedData$x.1,ContrivedData$x.2)

# Find the posterior density for the Contrived Data if all parameters were 1:
s.test <- krige.posterior(tau2=1,phi=1,sigma2=1,beta=rep(1,ncol(covariates)),
  y=ContrivedData$y,X=covariates,east=ContrivedData$s.1,north=ContrivedData$s.2)

# Print posterior density
s.test
```

---

lowerCombined — *State Legislative District (Lower Chambers) Public Opinion Ideology in 2010*

Description

These data present measures of ideology in 2010 for the districts for lower chambers of state legislatures, recorded as the variable krige.lower. 49 states' chambers are covered–the Nebraska Unicameral is omitted here to be included in the file upperCombined. Forecasts are based on a kriging model fitted over the 2008 Cooperative Congressional Election Survey (CCES), paired with predictive data from the 2010 Census. Each district's public ideology is paired with a measure of the ideology of the State House member (or members) from the district (update from Shor and McCarty 2011).

Format

The lowerCombined dataset has 5446 observations and 10 variables.

st Two-letter postal abbreviation for the state.

lower The state legislative district number (lower chamber).

STATEA The FIPS code for the state.

krige.lower The ideology of the average citizen in the district.

lowerKluge Combined index of STATEA followed by lower.

krige.lower.var The variance of ideology among the district's citizens.

name Last name of the state legislator, followed by first name and middle initial.

party Political party of the legislator. D=Democrat, R=Republican, X=Other.

st_id Temporary identifer variable. DO NOT USE.

np_score Ideology score for the state legislator (lower chamber). Higher values are usually interpreted as more right-wing, with lower values as more left-wing.

Source

Ansolabehere, Stephen. 2011. "CCES, Common Content, 2008." Ver. 4.

Minnesota Population Center. 2011. *National Historical Geographic Information System: Version 2.0.* Minneapolis, MN: University of Minnesota. 'https://www.nhgis.org'

Shor, Boris and Nolan M. McCarty. 2011. "The Ideological Mapping of American Legislatures." *American Political Science Review* 105(3):530-551.

References

Jeff Gill. 2020. Measuring Constituency Ideology Using Bayesian Universal Kriging. *State Politics & Policy Quarterly*. doi:10.1177/1532440020930197

Examples

```
# Descriptive Statistics
summary(lowerCombined)

# Correlate Senators' DW-NOMINATE Scores with Public Opinion Ideology
cor(lowerCombined$np_score,lowerCombined$krige.lower,use="complete.obs")

# Plot Legislators' DW-NOMINATE Scores against Public Opinion Ideology
plot(y=lowerCombined$np_score,x=lowerCombined$krige.lower,
     xlab="District Ideology (Kriging)", ylab="Legislator Ideology (Shor & McCarty)",
     main="State Legislatures: Lower Chambers", type="n")#
points(y=lowerCombined$np_score[lowerCombined$party=="R"],
       x=lowerCombined$krige.lower[lowerCombined$party=="R"],pch=".",col="red")
points(y=lowerCombined$np_score[lowerCombined$party=="D"],
       x=lowerCombined$krige.lower[lowerCombined$party=="D"],pch=".",col="blue")
```

mcmc.samples *Extract MCMC Samples*

Description

Extract MCMC samples estimated by metropolis.krige()

Usage

```
mcmc.samples(object, as.matrix, as.data.frame, ...)

## S3 method for class 'krige'
mcmc.samples(object, as.matrix = !as.data.frame, as.data.frame = FALSE, ...)

## S3 method for class 'summary.krige'
mcmc.samples(object, as.matrix = !as.data.frame, as.data.frame = FALSE, ...)

## S3 method for class 'krige'
as.matrix(x, ...)

## S3 method for class 'summary.krige'
as.matrix(x, ...)

## S3 method for class 'krige'
as.data.frame(x, ...)

## S3 method for class 'summary.krige'
as.data.frame(x, ...)
```

Arguments

object	A krigeor summary.krige object from the metropolis.krige function.
as.matrix	Logical values indicating if the output format should be a matrix. Defaults to TRUE.
as.data.frame	Logical values indicating if the output format should be a data.frame. Defaults to FALSE.
...	Additional arguments passed to as.matrix or as.data.frame methods.
x	A krige or summary.krige object for as.matrix and as.data.frame methods.

Details

The function extracts the MCMC samples from the a krigeor summary.krige object from the metropolis.krige function. Users can define the output by using as.matrix or as.data.frame.

Value

A summary.krige list object.

See Also

as.mcmc.krige

Examples

```
## Not run:
# Summarize Data
summary(ContrivedData)

# Initial OLS model
contrived.ols<-lm(y~x.1+x.2,data=ContrivedData)
# summary(contrived.ols)

# Set seed
set.seed(1241060320)

M <- 100
#M<-10000

contrived.run <- metropolis.krige(y ~ x.1 + x.2, coords = c("s.1","s.2"),
   data = ContrivedData, n.iter = M, n.burnin = 20, range.tol = 0.05)

contrived.run.mat <- mcmc.samples(contrived.run)

### Alternatively, use generic methods
contrived.run.mat <- as.matrix(contrived.run)
contrived.run.df <- as.data.frame(contrived.run)

## End(Not run)
```

---

metropolis.krige *Sampling Technique Using Metropolis-Hastings*

Description

This function performs Metropolis-Hastings sampling for a linear model specified over point-referenced geospatial data. It returns MCMC iterations, with which results of the geospatial linear model can be summarized.

Usage

```
metropolis.krige(
  formula,
  coords,
```

```
  data,
  n.iter = 100,
  powered.exp = 2,
  n.burnin = 0,
  y,
  X,
  east,
  north,
  na.action = "na.fail",
  spatial.share = 0.5,
  range.share = 0.5,
  beta.var = 10,
  range.tol = 0.05,
  b.tune = 1,
  nugget.tune = 10,
  psill.tune = 1,
  distance.matrix = FALSE,
  progress.bar = "message",
  accept.rate.warning = TRUE
)
```

Arguments

formula	An object of class formula (or one that can be coerced to that classes). A symbolic description of the model to be fitted. Alternatively, the model can be specified in y (a vector of the outcome variable) and X (a matrix of explanatory variables).
coords	A matrix of coordinates for all observations or a vector of variable names indicating the coordinates variables in the data. Alternatively, the coordinates can also be specified seperately using east and north.
data	An data frame containing the variables in the model.
n.iter	Number of MCMC iterations (defaults to 100).
powered.exp	This exponent, which must be greater than 0 and less than or equal to 2, specifies a powered exponential correlation structure for the data. One widely used specification is setting this to 1, which yields an exponential correlation structure. Another common specification is setting this to 2 (the default), which yields a Gaussian correlation structure.
n.burnin	Number of iterations that will be discarded for burnin (warmup). The number of burnin should not be larger than n.iter and the default is 0.
y	Alternative specification for the outcome variable that is used in the kriging model. If formula is used, this argument will be suppressed.
X	Alternative specification for the matrix of explanatory variables used in the kriging model. Different forms of the variables such as transformations and interactions also need to be specified accordingly beforehand.
east	Alternative specification for the vector of eastings for all observations.
north	Alternative specification for the vector of northing for all observations.

`na.action` A function which indicates what should happen when the data contain NAs. The default is "na.fail." Another possible value is "na.omit."

`spatial.share` Prior for proportion of unexplained variance that is spatial in nature. Must be greater than 0 and less than 1. Defaults to an even split, valued at 0.5.

`range.share` Prior for the effective range term, as a proportion of the maximum distance in the data. Users should choose the proportion of distance at which they think the spatial correlation will become negligible. Must be greater than 0. Values greater than 1 are permitted, but users should recognize that this implies that meaningful spatial correlation would persist outside of the convex hull of data. Defaults to half the maximum distance, valued at 0.5.

`beta.var` Prior for the variance on zero-meaned normal priors on the regression coefficients. Must be greater than 0. Defaults to 10.

`range.tol` Tolerance term for setting the effective range. At the distance where the spatial correlation drops below this term, it is judged that the effective range has been met. The default value is the commonly-used 0.05. Must be greater than 0 and less than 1.

`b.tune` Tuning parameter for candidate generation of regression coefficients that must be greater than 0. A value of 1 means that draws will be based on the variance-covariance matrix of coefficients from OLS. Larger steps are taken for values greater than 1, and smaller steps are taken for values from 0 to 1. Defaults to 1.0.

`nugget.tune` Tuning parameter for candidate generation of the nugget term (`tau2`) that must be greater than 0. A value of 1 means that draws will be based on the typical variance of an inverse gamma distribution. *Smaller* steps are taken for values *greater* than 1, and *larger* steps are taken for *decimal* values from 0 to 1. Defaults to 10.0.

`psill.tune` Tuning parameter for candidate generation of the partial sill term (`sigma2`) that must be greater than 0. A value of 1 means that draws will be based on the typical variance of an inverse gamma distribution. *Smaller* steps are taken for values *greater* than 1, and *larger* steps are taken for *decimal* values from 0 to 1. Defaults to 1.0.

`distance.matrix`
Logical value indicates whether to save the distance matrix in the output object. Saving distance matrix can save time for furthur use such as in `update()` function but may results in larger file size. Defaults to `FALSE`.

`progress.bar` Types of progress bar. The default is "message" and will report variance terms. Other possible values are "TRUE" (simple percentage) and "FALSE" (suppress the progress bar).

`accept.rate.warning`
Logical values indicating whether to show the warnings when the acceptance rates are too high or too low. Defaults to `TRUE`.

Details

Analysts should use this function if they want to estimate a linear regression model in which each observation can be located at points in geographic space. That is, each observation is observed for a set of coordinates in eastings & northings or longitude & latitude.

Researchers must specify their model in the following manner: formula should be a symbolic description of the model to be fitted; it is similar to R model syntax as used in lm(). In addition, a matrix of coordinates must be specified for the geospatial model in coords. coords should be a matrix with two columns that specify west-east and north-south coordinates, respectively (ideally eastings and northings but possibly longitude and latitude). It can also be a vector of strings indicating the variables names of the coordinates in the data. data should be a data frame containing the variables in the model including both the formula and coordinates (if only the names are provided). Alternatively, users can also specify the variables using y, X, east, and north for outcome, explanatory, west-east coordinates, and north-south coordinates variables, respectively. This alternative specification is compatible with the one used in the early version of this package.

n.iter is the number of iterations to sample from the posterior distribution using the Metropolis-Hastings algorithm. This defaults to 100 iterations, but many more iterations would normally be preferred. n.burnin is set to 0 by default to preserve all the iterations since the kriging model usually takes a relatively long time to run. Users can set a number for burnin or use burnin function afterwards to discard the burnin period. The output of the function prints the proportion of candidate values for the coefficients and for the variance terms accepted by the Metropolis-Hastings algorithm. Particularly low or high acceptance rates respectively may indicate slow mixing (requiring more iterations) or a transient state (leading to nonconvergence), so additional messages will print for extreme acceptance rates. Users may want to adjust the tuning parameters b.tune, nugget.tune, or psill.tune, or perhaps the tolerance parameter range.tol if the acceptance rate is too high or too low.

The function returns a "krige" list object including the output MCMC matrix of sampled values from the posterior distribution as well as the record of function arguments, model frame, acceptance rates, and standing parameters. Users can use the generic summary function to summarize the results or extract the elements of the object for further use.

Value

An object of class krige that includes the output MCMC matrix of sampled values from the posterior distribution as well as the record of function arguments, model frame, acceptance rates, and standing parameters.

References

Jeff Gill. 2020. Measuring Constituency Ideology Using Bayesian Universal Kriging. *State Politics & Policy Quarterly*. doi:10.1177/1532440020930197

Examples

```
## Not run:
# Summarize example data
summary(ContrivedData)

# Initial OLS model
contrived.ols<-lm(y~x.1+x.2,data=ContrivedData)
# summary(contrived.ols)

# Set seed
set.seed(1241060320)
```

```
#For simple illustration, we set to few iterations.
#In this case, a 10,000-iteration run converges to the true parameters.
#If you have considerable time and hardware, delete the # on the next line.
#10,000 iterations took 39 min. with 8 GB RAM & a 1.5 GHz Quad-Core processor.
M <- 100
#M<-10000

contrived.run <- metropolis.krige(y ~ x.1 + x.2, coords = c("s.1","s.2"),
   data = ContrivedData, n.iter = M, n.burnin=20, range.tol = 0.05)
# Alternatively, use burnin() after estimation
#contrived.run <- burnin(contrived.run, n.burnin=20)

# Summarize the results and examine results against true coefficients
summary(contrived.run)
(TRUTH<-c(0.5,2.5,0.5,0,1,2))

# Extract the MCMC matrix of the posterior distribution
contrived.run.mat <- mcmc.samples(contrived.run)
head(contrived.run.mat)

# Diagnostics
geweke(contrived.run, early.prop=0.5)
heidel.welch(contrived.run)

# Semivariogram
### Semivariance
semivariance(contrived.run)
### Plot semivariogram
semivariogram(contrived.run)
### Alternatively, use generic plot() on a krige object
plot(contrived.run)

## End(Not run)
```

`NY_subset` *New York State CCES Respondents in 2008*

Description

These data are a subset of the 2008 Cooperative Congressional Election Survey (CCES) Common Content. Only 1108 respondents from the state of New York are included, with predictors drawn from Gill's (2020) model of self-reported ideology. The CCES data are merged with predictors on geographic location based on ZIP codes (from ArcGIS & TomTom) and county ruralism (from the USDA).

Format

The `NY_subset` dataset has 1108 observations and 26 variables.

`state` The state abbreviation of the respondent's residence.

`zip` The respondent's ZIP code.

`age` The age of the respondent in years.

`female` An indicator of whether the respondent is female.

`ideology` The respondent's self-reported ideology on a scale of 0 (liberal) to 100 (conservative).

`educ` The respondent's level of education. 0=No Highschool, 1=High School Graduate, 2=Some College, 3=2-year Degree, 4=4-year degree, 5=Post-Graduate.

`race` The respondent's race. 1=White, 2=African American, 3=Nonwhite & nonblack.

`empstat` The respondent's employment status. 1=employed, 2=unemployed, 3=not in workforce.

`ownership` Indicator for whether the respondent owns his or her own home.

`inc14` The respondent's self reported income. 1=Less than $10,000, 2=$10,000-$14,999, 3=$15,000-$19,000, 4=$20,000-$24,999, 5=$25,000-$29,999, 6=$30,000-$39,999, 7=$40,000-$49,999, 8=$50,000-$59,999, 9=$60,000-$69,999, 10=$70,000-$79,999, 11=$80,000-$89,999, 12=$100,000-$119,999, 13=$120,000-$149,999, 14=$150,000 or more.

`catholic` Indicator for whether the respondent is Catholic.

`mormon` Indicator for whether the respondent is Mormon.

`orthodox` Indicator for whether the respondent is Orthodox Christian.

`jewish` Indicator for whether the respondent is Jewish.

`islam` Indicator for whether the respondent is Muslim.

`mainline` Indicator for whether the respondent is Mainline Christian.

`evangelical` Indicator for whether the respondent is Evangelical Christian.

`FIPS_Code` FIPS code of the repondent's state.

`rural` Nine-point USDA scale of the ruralism of each county, with 0 meaning the most urban and 8 meaning the most rural.

`zipPop` Indicates the population of the repondent's ZIP code.

`zipLandKM` Indicates the land area in square kilometers of the repondent's ZIP code.

`weight` Survey weights created by the CCES.

`cd` The congressional district the respondent resides in.

`fipsCD` Index that fuses the state FIPS code in the first two digits and the congressional district number in the last two digits.

`northings` Indicates the geographical location of the respondent in kilometer-based northings.

`eastings` Indicates the geographical location of the respondent in kilometer-based eastings.

Source

Ansolabehere, Stephen. 2011. "CCES, Common Content, 2008." Ver. 4.

ArcGIS. 2012. "USA ZIP Code Areas." https://www.arcgis.com/home/item.html?id=8d2012a2016e484dafaac0451f9

United States Department of Agriculture. 2013. "2013 Rural-Urban Continuum Codes." https://www.ers.usda.gov/data-products/rural-urban-continuum-codes.aspx

References

Jeff Gill. 2020. Measuring Constituency Ideology Using Bayesian Universal Kriging. *State Politics & Policy Quarterly*. doi:10.1177/1532440020930197

Examples

```
## Not run:
ny <- NY_subset

#data cleaning
ny$cathOrth<-ny$catholic+ny$orthodox
ny$consRelig<-ny$mormon+ny$evangelical
ny$jewMus<-ny$jewish+ny$islam

# Explanatory Variable Matrix
psrm.data <-cbind(ny$age, ny$educ, I(ny$age*ny$educ), as.numeric(ny$race==2),
       as.numeric(ny$race==3), ny$female, I(as.numeric(ny$race==2)*ny$female),
       I(as.numeric(ny$race==3)*ny$female), ny$cathOrth, ny$consRelig,
       ny$jewMus, ny$mainline, ny$rural, ny$ownership,
       as.numeric(ny$empstat==2), as.numeric(ny$empstat==3),ny$inc14)

dimnames(psrm.data)[[2]] <- c("Age", "Education", "Age.education",
                              "African.American", "Nonwhite.nonblack","Female",
                              "African.American.female", "Nonwhite.nonblack.female",
                              "Catholic.Orthodox", "Evang.Mormon", "Jewish.Muslim",
                              "Mainline","Ruralism", "Homeowner", "Unemployed",
                              "Not.in.workforce","Income")

# Outcome Variable
ideo <- matrix(ny$ideology,ncol=1)

# Set Number of Iterations:
# WARNING: 20 iterations is intensive on many machines.
# This example was tuned on Amazon Web Services (EC2) over many hours
# with 20,000 iterations--unsuitable in 2020 for most desktop machines.
#M<-20000
M<-100
set.seed(1,kind="Mersenne-Twister")

# Estimate the Model
ny.fit <- metropolis.krige(formula = ideo ~ psrm.data, coords = cbind(ny$eastings, ny$northings),
          powered.exp=1, n.iter=M, spatial.share=0.31,range.share=0.23,beta.var=10,
          range.tol=0.01, b.tune=0.1, nugget.tune=20, psill.tune=5)

# Discard first 20% of Iterations as Burn-In (User Discretion Advised).
ny.fit <- burnin(ny.fit, M/5)

# Summarize Results
summary(ny.fit)

#Convergence Diagnostics: Geweke and Heidelberger-Welch
geweke(ny.fit)
```

```
heidel.welch(ny.fit)

# Draw Semivariogram
semivariogram(ny.fit)

## End(Not run)
```

`NYcity_subset` *New York City CCES Respondents in 2008*

Description

These data are a subset of the 2008 Cooperative Congressional Election Survey (CCES) Common Content. Only 568 respondents from New York City are included, with predictors drawn from Gill's (2020) model of self-reported ideology. The CCES data are merged with predictors on geographic location based on ZIP codes (from ArcGIS & TomTom) and county ruralism (from the USDA).

Format

The `NYcity_subset` dataset has 568 observations and 26 variables.

`state` The state abbreviation of the respondent's residence.

`zip` The respondent's ZIP code.

`age` The age of the respondent in years.

`female` An indicator of whether the respondent is female.

`ideology` The respondent's self-reported ideology on a scale of 0 (liberal) to 100 (conservative).

`educ` The respondent's level of education. 0=No Highschool, 1=High School Graduate, 2=Some College, 3=2-year Degree, 4=4-year degree, 5=Post-Graduate.

`race` The respondent's race. 1=White, 2=African American, 3=Nonwhite & nonblack.

`empstat` The respondent's employment status. 1=employed, 2=unemployed, 3=not in workforce.

`ownership` Indicator for whether the respondent owns his or her own home.

`inc14` The respondent's self reported income. 1=Less than $10,000, 2=$10,000-$14,999, 3=$15,000-$19,000, 4=$20,000-$24,999, 5=$25,000-$29,999, 6=$30,000-$39,999, 7=$40,000-$49,999, 8=$50,000-$59,999, 9=$60,000-$69,999, 10=$70,000-$79,999, 11=$80,000-$89,999, 12=$100,000-$119,999, 13=$120,000-$149,999, 14=$150,000 or more.

`catholic` Indicator for whether the respondent is Catholic.

`mormon` Indicator for whether the respondent is Mormon.

`orthodox` Indicator for whether the respondent is Orthodox Christian.

`jewish` Indicator for whether the respondent is Jewish.

`islam` Indicator for whether the respondent is Muslim.

`mainline` Indicator for whether the respondent is Mainline Christian.

`evangelical` Indicator for whether the respondent is Evangelical Christian.

`FIPS_Code` FIPS code of the repondent's state.

`rural` Nine-point USDA scale of the ruralism of each county, with 0 meaning the most urban and 8 meaning the most rural.

`zipPop` Indicates the population of the repondent's ZIP code.

`zipLandKM` Indicates the land area in square kilometers of the repondent's ZIP code.

`weight` Survey weights created by the CCES.

`cd` The congressional district the respondent resides in.

`fipsCD` Index that fuses the state FIPS code in the first two digits and the congressional district number in the last two digits.

`northings` Indicates the geographical location of the respondent in kilometer-based northings.

`eastings` Indicates the geographical location of the respondent in kilometer-based eastings.

Source

Ansolabehere, Stephen. 2011. "CCES, Common Content, 2008." Ver. 4.

ArcGIS. 2012. "USA ZIP Code Areas." https://www.arcgis.com/home/item.html?id=8d2012a2016e484dafaac0451f9

United States Department of Agriculture. 2013. "2013 Rural-Urban Continuum Codes." https://www.ers.usda.gov/data-products/rural-urban-continuum-codes.aspx

References

Jeff Gill. 2020. Measuring Constituency Ideology Using Bayesian Universal Kriging. *State Politics & Policy Quarterly*. doi:10.1177/1532440020930197

Examples

```
## Not run:
nyc <- NYcity_subset

#data cleaning
nyc$cathOrth<-nyc$catholic+nyc$orthodox
nyc$consRelig<-nyc$mormon+nyc$evangelical
nyc$jewMus<-nyc$jewish+nyc$islam

# Explanatory Variable Matrix
psrm.data <-cbind(nyc$age, nyc$educ, I(nyc$age*nyc$educ), as.numeric(nyc$race==2),
       as.numeric(nyc$race==3), nyc$female, I(as.numeric(nyc$race==2)*nyc$female),
       I(as.numeric(nyc$race==3)*nyc$female), nyc$cathOrth, nyc$consRelig,
       nyc$jewMus, nyc$mainline, nyc$rural, nyc$ownership,
       as.numeric(nyc$empstat==2), as.numeric(nyc$empstat==3),nyc$inc14)

dimnames(psrm.data)[[2]] <- c("Age", "Education", "Age.education",
                              "African.American", "Nonwhite.nonblack","Female",
                              "African.American.female", "Nonwhite.nonblack.female",
                              "Catholic.Orthodox", "Evang.Mormon", "Jewish.Muslim",
                              "Mainline","Ruralism", "Homeowner", "Unemployed",
                              "Not.in.workforce","Income")
```

```
# Outcome Variable
ideo <- matrix(nyc$ideology,ncol=1)

# WARNING: This example was tuned on Amazon Web Services (EC2) over many hours
# with 150,000 iterations--a strain in 2020 for most desktop machines.
# A test with few iterations allows illustration.
#M<-150000
M<-150
set.seed(1,kind="Mersenne-Twister")

# Estimate the Model
nyc.fit <- metropolis.krige(formula = ideo ~ psrm.data, coords = cbind(nyc$eastings, nyc$northings),
          powered.exp=1, n.iter=M, spatial.share=0.31,range.share=0.23,beta.var=10,
          range.tol=0.01, b.tune=0.1, nugget.tune=20, psill.tune=5)

# Discard first 20% of Iterations as Burn-In (User Discretion Advised).
nyc.fit <- burnin(nyc.fit, M/5)

# Summarize Results
summary(nyc.fit)

#Convergence Diagnostics: Geweke and Heidelberger-Welch
geweke(nyc.fit)
heidel.welch(nyc.fit)

# Draw Semivariogram
semivariogram(nyc.fit)

## End(Not run)
```

`predict.krige` *Predictions by Kriging*

Description

This function uses the results of a model estimated by `metropolis.krige` to make kriging-based predictions.

Usage

```
## S3 method for class 'krige'
predict(object, newdata, credible = FALSE, new.X, new.east, new.north, ...)
```

Arguments

`object`	An `krige` object estimated by `metropolis.krige`.
`newdata`	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are produced. Alternatively, the new data can be specified using `new.X`, `new.east`, and `new.north`.

credible	If a credible interval on predictions is desired, a user may specify a proportion between 0 and 1 to indicate the interval probability. For example, a value of 0.9 would create a 90% credible interval. If NULL, then no credible interval will be produced.
new.X	The matrix of independent variables for observations to be predicted.
new.east	Vector of eastings for observations to be predicted.
new.north	Vector of northings for observations to be predicted.
...	Additional arguments passed to predict methods. Not supported for krige objects.

Details

Analysts should use this function if they want to make kriged predictions for observations at new locations or predict fitted values at original locations. To do this, researchers first must estimate a model using the metropolis.krige function.

After estimating the model, a krige object can provide information from the model fitted with metropolis.krige, including the original imput data, coordinates, and the model results themselves. The prediction will also use the same powered.exp. new.data can be specified for predicting the observations at new location. Otherwise, the fitted values for the original data and locations will be produced.

By default, the function uses median values of parameters to make a single point prediction for every kriged data point. However, if the uses specifies a probability with the credible option, then the function will determine the predictions for all iterations of the MCMC sample. The point estimates will then be a median of these predictions, and a credible interval will be returned based on percentiles. Note that estimating a credible interval is substantially more intensive computationally, but has the benefit of reporting uncertainty in predictions.

Value

An object of class matrix with one prediction per row. By default the matrix has one column, as only point predictions are returned. If the credible option is specified, there are three columns respectively indicating a point estimate (median prediction from MCMC), lower bound of the credible interval, and upper bound of the credible interval.

References

Jeff Gill. 2020. Measuring Constituency Ideology Using Bayesian Universal Kriging. *State Politics & Policy Quarterly*. doi:10.1177/1532440020930197

Examples

```
## Not run:
# Summarize Data
summary(ContrivedData)

# Initial OLS model
contrived.ols<-lm(y~x.1+x.2,data=ContrivedData)
# summary(contrived.ols)
```

```
# Set seed
set.seed(1241060320)

M <- 100
#M<-10000

contrived.run <- metropolis.krige(y ~ x.1 + x.2, coords = c("s.1","s.2"),
   data = ContrivedData, n.iter = M, range.tol = 0.05)

# Predict fitted values
predict(contrived.run)

# Predict new data
euler<-c(0.2,0.7)
archimedes<-c(0.3,0.1)
pythagoras<-c(0.1,0.4)
mathematicians<-rbind(euler,archimedes,pythagoras)
basel<-c(0.1,0.8)
sicily<-c(0.4,0.1)
samos<-c(0.1,0.4)
new.locations<-rbind(basel,sicily,samos)
newDf <- as.data.frame(cbind(mathematicians, new.locations))
colnames(newDf) <- c("x.1", "x.2", "s.1", "s.2")

# Make predictions from median parameter values:
(median.pred <- predict(contrived.run, newdata = newDf))

# Make predictions with 90\
(cred.pred <- predict(contrived.run, newdata = newDf, credible=0.9))

## End(Not run)
```

`semivariance` *Semivariance for Geospatial Data*

Description

This function computes the empirical semivariance for a spatially-distributed variable. Based on the user's chosen level of coarsening, the semivariance is presented for various distances.

Usage

```
semivariance(object, ...)

## S3 method for class 'krige'
semivariance(object, bins = 13, terms = "all", plot = FALSE, ...)

## S3 method for class 'lm'
```

```
semivariance(
  object,
  bins = 13,
  coords,
  terms = c("raw", "residual"),
  east,
  north,
  plot = FALSE,
  ...
)

## Default S3 method:
semivariance(object, bins = 13, coords, data, east, north, plot = FALSE, ...)
```

Arguments

object	An object for which the semivariance is desired. The object can be a krige object, a lm object, or a vector of variables (or variable names in the data).
...	Additional arguments passed to semivariance methods.
bins	Number of bins into which distances should be divided. The observed distances will be split at equal intervals, and the semivariance will be computed within each interval. Defaults to 13 intervals.
terms	A vector of strings specifies for which the semivariogram is created. Options are "raw" (the semivariogram for raw data), "residual" (the semivariogram for residuals from linear regression).
plot	Logical values indicates whether a graph of the empirical semivariogram should be presented with a run of the function. Default omits the plot and only returns semivariance values. See semivariogram for additional plotting functions.
coords	A matrix of coordinates for all observations or a vector of variable names indicating the coordinates variables in the data. Alternatively, the coordinates can also be specified separately using east and north.
east	Alternative specification for the vector of eastings for all observations.
north	Alternative specification for the vector of northing for all observations.
data	If object is a variable name, a data frame must be provided.

Details

Semivariance is equal to half of the variance of the difference in a variable's values at a given distance. That is, the semivariance is defined as: $\gamma(h) = 0.5 * E[X(s+h) - X(s)]^2$, where X is the variable of interest, s is a location, and h is the distance from s to another location.

The function can be applied to a variable, a fitted linear model (lm object) before fitting a spatial model or to a krige object or semivariance object to assess the model fit. When applying to a variable, it will describes the raw data; for a lm object, the default will present empirical semivariogram for both the raw data and linear residuals. Users can also specify which semivariance is needed in the terms argument if there are multiple kinds of semivariogram can be plotted. A semivariance object can also be used to create semivariogram afterwards using generic plot function with more options.

Value

A semivariance object. It will be a numeric vector with each bin's value of the semivariance if only one kind of semivariance is computed; a list including different kinds of semivariance if both raw and residual semivariance is computed.

References

Sudipto Banerjee, Bradley P. Carlin, and Alan E. Gelfand. 2015. *Hierarchical Modeling and Analysis for Spatial Data*. 2nd ed. Boca Raton, FL: CRC Press.

See Also

semivariogram, plot.semivariance, exponential.semivariance

Examples

```
## Not run:
# Summarize example data
summary(ContrivedData)

# Empirical semivariance for variable y
semivariance(ContrivedData$y,coords = cbind(ContrivedData$s.1, ContrivedData$s.2))

# Initial OLS Model
contrived.ols<-lm(y~x.1+x.2,data=ContrivedData); summary(contrived.ols)

# Empirical semivariance for ols fit
(sv.ols <- semivariance(contrived.ols, coords = c("s.1","s.2"), bins=13))
plot(sv.ols)

# Estimation using metropolis.krige()
# Set seed
set.seed(1241060320)

M <- 100

contrived.run <- metropolis.krige(y ~ x.1 + x.2, coords = c("s.1","s.2"),
  data = ContrivedData, n.iter = M, range.tol = 0.05)

# Parametric semivariance
(sv.krige <- semivariance(contrived.run, plot = TRUE))

# Convert to other format for further use
as.matrix(sv.krige)
as.data.frame(sv.krige)

## End(Not run)
```

`semivariogram` *Semivariogram for Geospatial Data*

Description

This function creates semivariogram plots. It creates empirical semivariogram for raw data and `lm` object or parametric exponential semivariogram based on the estimation from `metropolis.krige`. Based on the user's chosen level of coarsening, the semivariogram is presented for various distances.

Usage

```
semivariogram(x, ...)

## S3 method for class 'krige'
semivariogram(x, ..., bins = 13, terms = "all", type, pch, lty, legend, col)

## S3 method for class 'krige'
plot(...)

## S3 method for class 'lm'
semivariogram(
  x,
  ...,
  coords,
  bins = 13,
  terms = c("raw", "residual"),
  east,
  north,
  type,
  legend,
  col,
  pch,
  lty
)

## Default S3 method:
semivariogram(
  x,
  ...,
  coords,
  data,
  bins = 13,
  east,
  north,
  type,
  pch,
  lty,
```

```
  col
)

## S3 method for class 'semivariance'
semivariogram(x, ..., type, pch, lty, legend, col)

## S3 method for class 'semivariance'
plot(x, ..., type, pch, lty, legend, col)
```

Arguments

x	An object for which a semivariogram is desired. The object can be a krige object, a semivariance object, a lm object, or a vector of variables (or variable names in the data).
...	Additional arguments to be passed to semivariogram methods. Further arguments that can passed to plot() function can be specified here.
bins	Number of bins into which distances should be divided. The observed distances will be split at equal intervals, and the semivariance will be computed within each interval. Defaults to 13 intervals.
terms	A vector of strings specifies for which the semivariogram is created. Options are "raw" (the semivariogram for raw data), "residual" (the semivariogram for residuals from linear regression). "parametric" (the parametric powered exponential semivariogram based on the estimation from metropolis.krige). Defaults will create all the applicable plots.
type	A vector specify the type of plots for each term. Options are "l" (line plot) and "p" (scatter plot). Defaults to c(raw = "p", residual = "p",parametric = "l")
pch	A vector specify the points symbols for scatter plot. Suppressed for line plot.
lty	A vector specify the line type for line plot. Suppressed for scatter plot.
legend	A logical argument for whether legend should be presented. Defaults to TRUE.
col	A vector specify the color for each term. Defaults to c(raw = "black",residual = "blue", parametric = "red")
coords	A matrix of coordinates for all observations or a vector of variable names indicating the coordinates variables in the data. Alternatively, the coordinates can also be specified separately using east and north.
east	Alternative specification for the vector of eastings for all observations.
north	Alternative specification for the vector of northing for all observations.
data	If object is a variable name, a data frame must be provided.

Details

The function creates semivariograms for diagnosing the spatial relationship that best describes the data and for diagnosing the degree to which the model fits the spatial relationship. With a view of the empirical semivariogram, a user can consult images of parametric semivariograms to determine whether an exponential, Gaussian, or other powered expoential function fit the data well, or if another style of semivariogram works better. Examining this also allows the user to develop priors

such as the approximate split in variance between the nugget and partial sill as well as the approximate distance of the effective range. Semivariograms are explicitly tied to a corresponding spatial correlation function, so determining the former automatically implies the latter. See Banerjee, Carlin, and Gelfand for a fuller explanation, as well as a guidebook to semivariogram diagnosis (2015, 26-30).

The function can be applied to a variable, a fitted linear model (`lm` object) before fitting a spatial model or to a `krige` object or `semivariance` object to assess the model fit. When applying to a variable, it will describes the raw data; for a `lm` object, the default will present empirical semivariogram for both the raw data and linear residuals; when applying to a `krige` object, the default will present empirical semivariogram for the raw data and the residuals from linear fit, and the parametric semivariogram given the estimates from the geospatial model fitted in `metropolis.krige`; for a `semivariance` object, it will present a plot(s) for whichever the semivariance is calculated. Users can also specify which semivariogram is needed in the `terms` argument if there are multiple kinds of semivariogram can be plotted. The plots are compatible to the arguments used in base `R` base graphics.

Value

An semivariogram plot. For `krige` object, it will return empirical semivariograms for raw data and residuals of linear regression as well as a parametric powered exponential semivariogram given the values of the estimates from `metropolis.krige` as default.

References

Sudipto Banerjee, Bradley P. Carlin, and Alan E. Gelfand. 2015. *Hierarchical Modeling and Analysis for Spatial Data*. 2nd ed. Boca Raton, FL: CRC Press.

See Also

`semivariance`, `exponential.semivariance`

Examples

```
## Not run:
# Summarize Data
summary(ContrivedData)

# Empirical semivariagram for variable y
semivariogram(x=ContrivedData$y, coords = cbind(ContrivedData$s.1, ContrivedData$s.2))

# Initial OLS Model
contrived.ols<-lm(y~x.1+x.2,data=ContrivedData)

# Empirical semivariagram for ols fit
semivariogram(contrived.ols, coords = c("s.1","s.2"), bins=13)

# Set seed
set.seed(1241060320)

M <- 100
#M<-10000
```

```
contrived.run <- metropolis.krige(y ~ x.1 + x.2, coords = c("s.1","s.2"),
   data = ContrivedData, n.iter = M, range.tol = 0.05)

# Parametric semivariagram
semivariogram(contrived.run, bins=13, terms = c("raw", "residual", "parametric"),
  type= c(raw = "p", residual = "p", parametric = "l"), legend = TRUE, col = c("black",
  "blue", "red"), pch = c(1,3,NA), lty = c(NA,NA,1))

# Alternatively, the generic function plot can also be applied to krige object
plot(contrived.run)

# Plot semivariance object
plot(semivariance(contrived.run, bins=13))

## End(Not run)
```

`stateCombined` *State Public Opinion Ideology in 2010*

Description

These data present measures of ideology in 2010 for the 50 American states, recorded as the variable `krige.state`. Forecasts are based on a kriging model fitted over the 2008 Cooperative Congressional Election Survey (CCES), paired with predictive data from the 2010 Census. Each state is listed twice, as each state's public ideology is paired with the DW-NOMINATE common space score of each of its two senators in 2011 (update from McCarty, Poole and Rosenthal 1997).

Format

The `stateCombined` dataset has 100 observations (2 each for 50 states) and 13 variables.

`STATEA` The FIPS code for the state.

`krige.state` The ideology of the average citizen in the state.

`krige.state.var` The variance of ideology among the state's citizens.

`cong` The term of Congress studied–112 for this dataset.

`idno` Identification number for the senator–ICPSR numbers continued by Poole & Rosenthal.

`state` The ICPSR code for the state.

`cd` The congressional district number–0 for senators.

`statenm` The first seven letters of the state's name.

`party` Political party of the senator. 100=Democrat, 200=Republican, 328=Independent.

`name` Last name of the senator, followed by first name if ambiguous.

`dwnom1` First dimension DW-NOMINATE common space score for the senator. Higher values are usually interpreted as more right-wing, with lower values as more left-wing.

`stateCD` Combined index of `STATEA` followed by `cd`.

`obama` Barack Obama's percentage of the two-party vote in the state in 2012.

Source

Ansolabehere, Stephen. 2011. "CCES, Common Content, 2008." Ver. 4.

McCarty, Nolan M., Keith T. Poole and Howard Rosenthal. 1997. *Income Redistribution and the Realignment of American Politics*. American Enterprise Institude Studies on Understanding Economic Inequality. Washington: AEI Press.

Minnesota Population Center. 2011. *National Historical Geographic Information System: Version 2.0.* Minneapolis, MN: University of Minnesota. 'https://www.nhgis.org'

References

Jeff Gill. 2020. Measuring Constituency Ideology Using Bayesian Universal Kriging. *State Politics & Policy Quarterly*. doi:10.1177/1532440020930197

Examples

```
# Descriptive Statistics
summary(stateCombined)

# Correlate Senators' DW-NOMINATE Scores with Public Opinion Ideology
cor(stateCombined$krige.state,stateCombined$dwnom1)

# Plot Senators' DW-NOMINATE Scores against Public Opinion Ideology
plot(y=stateCombined$dwnom1,x=stateCombined$krige.state,
    xlab="State Ideology (Kriging)", ylab="Legislator Ideology (1st Dim., Common Space)",
     main="U.S. Senate", type="n")
points(y=stateCombined$dwnom1[stateCombined$party==200],
       x=stateCombined$krige.state[stateCombined$party==200],pch="R",col="red")
points(y=stateCombined$dwnom1[stateCombined$party==100],
       x=stateCombined$krige.state[stateCombined$party==100],pch="D",col="blue")
```

`summary.krige` *Summarize Fitted Kriging Model*

Description

Create a summary of a estimated model from `metropolis.krige`

Usage

```
## S3 method for class 'krige'
summary(object, ...)
```

Arguments

`object`	An `krige` object from the `metropolis.krige` function.
`...`	Additional arguments passed to `summary` methods. Not supported for `krige` object.

Details

The function creates a summary of the model estimated by `metropolis.krige`. The output includes both the parameters and estimates of the model.

Value

A `summary.krige` list object.

See Also

`as.mcmc.summary.krige`

Examples

```
## Not run:
# Summarize data
summary(ContrivedData)

# Initial OLS model
contrived.ols<-lm(y~x.1+x.2,data=ContrivedData)
# summary(contrived.ols)

# Set seed
set.seed(1241060320)

#For simple illustration, we set to few iterations.
#In this case, a 10,000-iteration run converges to the true parameters.
#If you have considerable time and hardware, delete the # on the next line.
#10,000 iterations took 39 min. with 8 GB RAM & a 1.5 GHz Quad-Core processor.
M <- 100
#M<-10000

contrived.run <- metropolis.krige(y ~ x.1 + x.2, coords = c("s.1","s.2"),
   data = ContrivedData, n.iter = M, range.tol = 0.05)

# Summary
summary(contrived.run)

## End(Not run)
```

`update.krige` *Update Model*

Description

Update the Markov chain associated with the `metropolis.krige` model

Usage

```
## S3 method for class 'krige'
update(object, n.iter, n.burnin = 0, combine = FALSE, ...)
```

Arguments

object	An krige object from the metropolis.krige function.
n.iter	Number of iterations for the update run.
n.burnin	The number of burnin iterations. Defaults to 0.
combine	Logical value indicate whether to combine the update run with original output.
...	Additional arguments passed to update methods. Not supported for krige objects.

Details

Since MCMC calculations typically need to run relatively long. This function can continue the MCMC calculations by metropolis.krige(). The parameters of the original model and the estimated results from the previous run are passed through the krige object.

As geospatial estimation can be computationally taxing, the users may want to preserve more iterations of the posterior samples. The combine argument can be used to indicate whether combine the updated run with previous results. This includes both the posterior samples and acceptance rates.

Value

An object of class krige that includes the output MCMC matrix of sampled values from the posterior distribution as well as the record of function arguments, model frame, acceptance rates, and standing parameters.

Examples

```
## Not run:
# Summarize Data
summary(ContrivedData)

# Set seed
set.seed(1241060320)

M <- 100
#M<-10000

contrived.run <- metropolis.krige(y ~ x.1 + x.2, coords = c("s.1","s.2"),
   data = ContrivedData, n.iter = M, range.tol = 0.05)
summary(contrived.run)

# Update run
contrived.run2 <- update(contrived.run, n.iter = M, combine = TRUE)

summary(contrived.run2)
#plot(contrived.run2)
```

```
## End(Not run)
```

---

upperCombined *State Legislative District (Upper Chambers) Public Opinion Ideology in 2010*

Description

These data present measures of ideology in 2010 for the districts for upper chambers of state legislatures, recorded as the variable `krige.upper`. All 50 states' chambers are covered (including the Nebraska Unicameral). Forecasts are based on a kriging model fitted over the 2008 Cooperative Congressional Election Survey (CCES), paired with predictive data from the 2010 Census. Each district's public ideology is paired with a measure of the ideology of the State Senate member from the district (update from Shor and McCarty 2011).

Format

The `upperCombined` dataset has 1989 observations and 10 variables.

`st` Two-letter postal abbreviation for the state.

`upper` The state legislative district number (upper chamber).

`STATEA` The FIPS code for the state.

`krige.upper` The ideology of the average citizen in the district.

`upperKluge` Combined index of `STATEA` followed by `upper`.

`krige.upper.var` The variance of ideology among the district's citizens.

`name` Last name of the state legislator, followed by first name and middle initial.

`party` Political party of the legislator. D=Democrat, R=Republican, X=Other.

`st_id` Temporary identifer variable. DO NOT USE.

`np_score` Ideology score for the state legislator (upper chamber). Higher values are usually interpreted as more right-wing, with lower values as more left-wing.

Source

Ansolabehere, Stephen. 2011. "CCES, Common Content, 2008." Ver. 4.

Minnesota Population Center. 2011. *National Historical Geographic Information System: Version 2.0.* Minneapolis, MN: University of Minnesota. 'https://www.nhgis.org'

Shor, Boris and Nolan M. McCarty. 2011. "The Ideological Mapping of American Legislatures." *American Political Science Review* 105(3):530-551.

References

Jeff Gill. 2020. Measuring Constituency Ideology Using Bayesian Universal Kriging. *State Politics & Policy Quarterly*. doi:10.1177/1532440020930197

Examples

```
# Descriptive Statistics
summary(upperCombined)

# Correlate Senators' DW-NOMINATE Scores with Public Opinion Ideology
cor(upperCombined$np_score,upperCombined$krige.upper,use="complete.obs")

# Plot Legislators' DW-NOMINATE Scores against Public Opinion Ideology
plot(y=upperCombined$np_score,x=upperCombined$krige.upper,
     xlab="District Ideology (Kriging)", ylab="Legislator Ideology (Shor & McCarty)",
          main="State Legislatures: Upper Chambers", type="n")
points(y=upperCombined$np_score[upperCombined$party=="R"],
       x=upperCombined$krige.upper[upperCombined$party=="R"],pch=".",col="red")
points(y=upperCombined$np_score[upperCombined$party=="D"],
       x=upperCombined$krige.upper[upperCombined$party=="D"],pch=".",col="blue")
```

WVwells *West Virginia Oil and Gas Production in 2012*

Description

These data are a subset of the West Virginia Geological and Economic Survey of 2014. They contain information on the coordinates of wells that yielded at least some quantity of natural gas in 2012. In addition to coordinates, the data contain information on well ownership and operation, rock pressure at the well, elevation of the well, oil production, and gas production.

Format

The `WVwells` dataset has 1949 observations and 18 variables.

`APINum` A 10-digit number in the format assigned by the American Petroleum Institute (API), consisting of a 2-digit state code, a 3-digit county code with leading zeroes, and a 5-digit permit number with leading zeroes. Data Source: West Virginia Department of Environmental Protection, Office of Oil & Gas (WVDEP-OOG).

`CntyCode` A 3-digit numeric code, assigned in numeric order by county name. Data Source: The county code for a well is assigned by WVDEP-OOG, based on the well location.

`CntyName` The name of the county. Please see CntyCode (County Code) for a list of all West Virginia county names. Data Source: The county code for a well is assigned by WVDEP-OOG, based on the well location. The county name is a translation of the county code.

`Operator` The name of the operator who owns the well at the time of reporting. Data Source: WVDEP-OOG plat; verified on the WR-35 completion record.

`SurfaceOwn` The name of the owner of the surface property on which the well is located. Data Source: WVDEP-OOG plat; verified on the WR-35 completion record.

`MineralOwn` Mineral Owner: The name of the owner of the mineral rights where the well is located. Data Source: WVDEP-OOG plat.

`CompanyNum` The operator's serial number for the well. Data Source: WVDEP-OOG plat; verified on the WR-35 completion record.

`WellNum` The operator's number for the well on the surface property (farm). Data Source: WVDEP-OOG plat; verified on the WR-35 completion record.

`UTMESrf` Surface Location–Universal Transverse Mercator, Easting: The well location at the surface measured in meters to one decimal point, east of the central meridian in UTM Zone 17; datum: NAD83. Data Source: Taken directly from the plat if given as such. Otherwise, computed from the location reported on the plat. Suspect locations may be adjusted using various additional resources (e.g. topographic maps) if deemed necessary.

`UTMNSrf` Surface Location–Universal Transverse Mercator, Northing: The well location at the surface measured in meters to one decimal point, north of the equator in UTM Zone 17; datum: NAD83. Data Source: Taken directly from the plat if given as such. Otherwise, computed from the location reported on the plat. Suspect locations may be adjusted using various additional resources (e.g. topographic maps) if deemed necessary.

`LonSrf` Surface Location–Longitude: The well location at the surface measured to a precision of 6 decimal points, in degrees west of the Prime Meridian. Data Source: Taken directly from the plat if given as such. Otherwise, computed from the location reported on the plat. Suspect locations may be adjusted using various additional resources (e.g. topographic maps) if deemed necessary.

`LatSrf` Surface Location–Latitude: The well location at the surface measured to a precision of 6 decimal points, in degrees north of the equator. Data Source: Taken directly from the plat if given as such. Otherwise, computed from the location reported on the plat. Suspect locations may be adjusted using various additional resources (e.g. topographic maps) if deemed necessary.

`Elevation` Elevation: The height of the well in feet above mean sea level. Data Source: WVDEP-OOG plat; verified on the WR-35 completion record.

`RockPres` Formation Rock Pressure at Surface: The pressure measured at the surface usually after stimulation, in pounds per square inch (psi). Data Source: WVDEP-OOG WR-35 comp#' letion record, submitted by the operator to WVDEP-OOG.

`GProd2012` 2012 Gas Production Reported: The total gas production for the well for 2012 in thousands of cubic feet (MCF); includes all pay zones. Data Source: Production data reported by the operator to the State regulatory authority for Oil and Gas (WVDEP-OOG); WVGES obtained the data from WVDEP-OOG.

`OProd2012` 2012 Oil Production Reported: The total oil production for the well for 2012 in barrels (Bbl); includes all pay zones. Production data reported by the operator to the State regulatory authority for Oil and Gas (WVDEP-OOG); WVGES obtained the data from WVDEP-OOG.

`logElevation` Logarithm of `Elevation`.

Source

West Virginia Geological and Economic Survey. 2014. "WVMarcellusWellsCompleted102014." Morgantown, WV. http://www.wvgs.wvnet.edu/www/datastat/devshales.htm Accessed via: FracTracker. 2019. "West Virginia Oil & Gas Activity." https://www.fractracker.org/map/us/west-virginia/

References

Jason S. Byers & Jeff Gill. N.D. "Applied Geospatial Data Modeling in the Big Data Era: Challenges and Solutions."

Examples

```
## Not run:
# Descriptive Statistics
summary(WVwells)

# Record means of predictors:
# These are used BOTH to eliminate the intercept and to recover predictions later.
mean.logGas<-mean(WVwells$logGProd2012);mean.logGas
mean.logElevation<-mean(WVwells$logElevation);mean.logElevation
mean.RockPres<-mean(WVwells$RockPres);mean.RockPres

# Outcome Variable, De-Meaned
WVwells$logGas <- WVwells$logGProd2012-mean.logGas

# Explanatory Variable: DE-MEANED PREDICTORS AND NO CONSTANT TERM
# Because we deducted the mean from all predictors and the outcome,
# it is valid to do regression through the origin.
WVwells$LogElevation <- WVwells$logElevation-mean.logElevation
WVwells$RockPressure <- WVwells$RockPres-mean.RockPres

# OLS Model
fracking.ols<-lm(logGas~LogElevation+RockPressure-1, data = WVwells)
summary(fracking.ols)

intercept.mod<-lm(logGProd2012~ logElevation+RockPres,data=WVwells)
summary(intercept.mod)

# Set Number of Iterations:
# WARNING: 100 iterations is intensive on many machines.
# This example was tuned on Amazon Web Services (EC2) over many hours
# with 5,000 iterations--unsuitable in 2020 for most desktop machines.
#M<-5000
M<-20

set.seed(1000, kind="Mersenne-Twister")#SET SEED FOR CONSISTENCY

# Trial Run, Linear Model of Ideology with Geospatial Errors Using Metropolis-Hastings:
wv.fit <- metropolis.krige(logGas~LogElevation+RockPressure-1, coords = c("UTMESrf", "UTMNSrf"),
                           data = WVwells, n.iter=M, powered.exp=0.5, spatial.share=0.60,
                            range.share=0.31, beta.var=1000, range.tol=0.1, b.tune=1,
                            nugget.tune=1, psill.tune=30)

# Discard first 20% of Iterations as Burn-In (User Discretion Advised).
wv.fit <- burnin(wv.fit, M/5)

# Summarize Results
summary(wv.fit)
```

```
# Convergence Diagnostics
# geweke(wv.fit) Not applicable due to few iterations.
heidel.welch(wv.fit)

# Draw Semivariogram
semivariogram(wv.fit)

# Predictive Data for Two Wells Tapped in 2013
well.1<-c(log(1110)-mean.logElevation,1020-mean.RockPres)
well.2<-c(log(643)-mean.logElevation,630-mean.RockPres)
site.1<-c(557306.0, 4345265)
site.2<-c(434515.7, 4258449)
well.newdata <- as.data.frame(cbind(rbind(well.1,well.2),rbind(site.1,site.2)))
colnames(well.newdata)<-c("LogElevation", "RockPressure", "UTMESrf","UTMNSrf")

# Make predictions from median parameter values:
(median.pred <- predict(wv.fit, newdata = well.newdata))

# Prediction in thousands of cubic feet (MCF):
round(exp(median.pred+mean.logGas))

# Make predictions with 90\
(cred.pred <- predict(wv.fit, newdata = well.newdata, credible = 0.9))

# Prediction in thousands of cubic feet (MCF) and the true yield in 2013:
Actual.Yield<-c(471171, 7211)
round(cbind(exp(cred.pred+mean.logGas),Actual.Yield))

## End(Not run)
```

Index

∗ **data**
congCombined, 5
ContrivedData, 7
lowerCombined, 15
NY_subset, 22
NYcity_subset, 25
stateCombined, 35
upperCombined, 39
WVwells, 40

as.data.frame.krige *(*mcmc.samples*)*, 17
as.data.frame.semivariance *(*as.matrix.semivariance*)*, 2
as.data.frame.summary.krige *(*mcmc.samples*)*, 17
as.matrix.krige *(*mcmc.samples*)*, 17
as.matrix.semivariance, 2
as.matrix.summary.krige *(*mcmc.samples*)*, 17
as.mcmc.krige, 3, *18*
as.mcmc.summary.krige, *37*
as.mcmc.summary.krige *(*as.mcmc.krige*)*, 3

burnin, 5

coda::as.mcmc(), *4*
congCombined, 5
ContrivedData, 7

exponential.semivariance, 8, *9*, *31*, *34*

formula, *19*

geweke, 10, *12*
geweke.krige, *11*
geweke.summary.krige, *11*

heidel.welch, 11
heidel.welch.krige, *12*
heidel.welch.summary.krige, *12*

krige, 13
krige.posterior, 13

lowerCombined, 15

mcmc.samples, 17
metropolis.krige, 18

NY_subset, 22
NYcity_subset, 25

plot.krige *(*semivariogram*)*, 32
plot.semivariance, *9*, *31*
plot.semivariance *(*semivariogram*)*, 32
predict.krige, 27

semivariance, 29, *34*
semivariogram, *9*, *30*, *31*, 32
stateCombined, 35
summary.krige, 36

update.krige, 37
upperCombined, 39

WVwells, 40