Package 'hydroMap'

August 29, 2018

Title hydroMap

Version 0.1

Description Interpolates groundwater level observations giving realistic flow lines.

Depends R (>= 3.2.2)

License GPL3 or later.

Encoding UTF-8

LazyData true

Imports sp, gstat, raster, parallel, rgenoud

Suggests RSAGA

RoxygenNote 6.0.1.9000

Author Tim Peterson [aut, cre]

Maintainer Tim Peterson <timjp@unimelb.edu.au>

R topics documented:

`import.DEM` *Reads in an ARCMAP ASCII grid file containing the DEM and prepares it for use in the head mapping.*

Description

`import.DEM` reading in a an ARCMAP ASCII grid file.

Usage

```
import.DEM(grid, debug.level = 0)
```

Arguments

grid	character string for the ARCMAP ASCII grid file name.
debug.level	Control the user messages. A value >0 outputs progress.

Details

This function reads in an ARCMAP ASCII grid file containing the DEM and prepares it for use in the mapping of groundwater level data. The preparations include removing NA grid cells and converting it to a fully gridded Spatial object. Also, the coordinates must be projected.

Value

A gridded Spatial object will be returned with one field named "DEM".

---

import.pointData *Reads in the point groundwater level observations file.*

Description

`import.pointData` reads in a .csv file of the point groundwater level data.

Usage

```
import.pointData(data, min.sepDist = 5, orderby.head = TRUE,
  debug.level = 0)
```

Arguments

data	character string for the point data .csv filename. The file must contain columns with the following names "easting", "northing" and "head".
min.sepDist	scalar value >0 defining the maximum distance between two observations that will result in a duplicate being detected. When a duplicate is detected then when `orderby.head==TRUE`, the point with the greatest water level elevation will be kept ALso, the lowest elevation point will be kept. The default is 5m. ' @param `orderby.head` logical scalar defining if the highest elevation water level should be selected when a duplicate is found. The default is `TRUE`.
debug.level	Control the user messages. A value >0 outputs progress.

Details

This function reads in .csv file of the point groundwater level data and removing duplicates points. The coordinates must be projected.

Value

A point Spatial object will be returned.

`krige.head` *Spatially interpolates sparse groundwater level observations, and if required, estimate mapping parameters.*

Description

`krige.head` creates a groundwater level elevation map from sparse point observations for one time point.

Usage

```
krige.head(formula = as.formula("head ~ elev + MrVBF + MrRTF + smoothing"),
  grid = NULL, grid.landtype.colname = NULL, data = NULL,
  data.fixedHead = NULL, newdata = NULL, data.errvar.colname = NULL,
  model = NULL, mrvbf.pslope = if (any(match(all.vars(as.formula(formula)),
  "MrVBF", nomatch = F) | match(all.vars(as.formula(formula)), "MrRTF", nomatch
  = F))) {     seq(0.5, 1.5, length.out = 11) } else {     NULL },
  mrvbf.ppctl = if (any(match(all.vars(as.formula(formula)), "MrVBF", nomatch
  = F) | match(all.vars(as.formula(formula)), "MrRTF", nomatch = F))) {
  seq(0.5, 1.5, length.out = 11) } else {     NULL }, smooth.std = seq(0.5,
  1.5, length.out = 11), nmax = if (is.character(data)) {     -999 } else {
   ceiling(seq(0.1, 0.2, 0.01) * length(data)) }, nmax.fixedHead = if
  (!is.null(data.fixedHead)) {     seq(10, 110, length = 11) } else {     NULL
  }, maxdist = if (class(grid) == "SpatialPixelsDataFrame" || class(grid) ==
  "SpatialGridDataFrame") {     ceiling(0.5 * sqrt((extent(grid)[2] -
  extent(grid)[1])^2 + (extent(grid)[4] - extent(grid)[3])^2) * seq(0.1, 1,
  0.1)) } else {     -999 }, trendMaxDistFrac = seq(0.1, 1, 0.1),
  nmin = 0.2, omax = NULL, nsim = 0, fit.variogram.type = 1,
  objFunc.type = 1, use.cluster = TRUE, debug.level = 0, ...)
```

Arguments

`formula`	defines the R formula (as a character or formula data type) to be used to interpolate the heads. The left hand side of the formula must be `head`. The right hand side can contain any or all of the following terms: `elev` for the land surface elevation; `MrVBF` for the Multiresolution Index of Valley Bottom Flatness as a measure of valley-ness at each DEM grd cell; `MrRTF` for the Multiresolution Index of Ridge Top Flatness as a measure of ridge to plateaus at each DEM grd cell; `smoothing` for a local smoothing factor derived from the DEM roughness. For any terms other than the prior, the data for the variable must be listed within the inputs `grid` and `data`. The default is `as.formula("head ~ elev + MrVBF + MrRTF + smoothing")`.

grid is either a character string to a ASCII grid digital elevation model (DEM) or a SpatialPixelsDataFrame or SpatialGridDataFrame containing the land surface elevation, which must be named elev, and each formula variable other than MrVBF, MrRTF and smoothing; which are each derived from the DEM.

grid.landtype.colname
is a character of the column name within grid and data that define the land category. The land category data should be an integer and be of a sufficiently small number of categories that multiple data points exists within each land category. If NULL, then land categories are not accounted for in the mapping. The default is NULL.

data is either a character string to a .csv file of point data or a SpatialPointsDataFrame containing the columns Easting, Northing and head or depth. If the formula includes the term elev, then the bore elevation should included in data so as to account for difference between the bore and DEM elevation. Each formula right hand side variable other than MrVBF, MrRTF and smoothing should also provided within data.

data.fixedHead is as for data but the points are treated as fixed head points within by a cokriging approach. It can be used to guide the head estimates toward zero along, say, the coastline. The fixed points will have a greater influence when no observation data is nearby. If NULL, then no fixed head points are used. The default is NULL.

newdata is as for data but the points in a split-sample cross-validation scheme to estimate the interpolation error. Points listed within newdata should not be listed within data. If is.null(newdata)==TRUE, then grid should be NULL.

data.errvar.colname
is a character of the column name within data that define measurement error, as a variance. If NULL, then measurement error is not accounted for. The default is NULL.

model is either a character for the name of the variogram model type or a gstat variogram model object of type variogramModel. The available options are as per gstat, but it is suggested to be Mat.

mrvbf.pslope defines the MrVFB shape parameter for the slope (see Gallant et al. 2003), a vector of two values defining the optimisation range when the parameter is treated as a real number or a vector of length >2 values defining the optimisation increments when the parameter is treated as not continuous but discrete. If a single number is input, then the parameter will not be optimised. If the formula includes either of the terms MrVBF or MrRTF, then the default is seq(0.5, 1.5, length.out = 11). Else, the default is NULL.

mrvbf.ppctl defines the MrVFB shape parameter for elevation percentile (see Gallant et al. 2003). It can be a scalar number, a vector of two values defining the optimisation range when the parameter is treated as a real number or a vector of length >2 values defining the optimisation increments when the parameter is treated as not continuous but discrete. If a single number is input, then the parameter will not be optimised. If the formula includes either of the terms MrVBF or MrRTF, then the default is seq(0.5, 1.5, length.out = 11). Else, the default is NULL.

smooth.std defines the strength of the Gaussian kernal smoothing applied to the 5x5 grid cells surrounding each DEM grid cell. It can be a scalar number, a vector of

two values defining the optimisation range when the parameter is treated as a real number or a vector of length >2 values defining the optimisation increments when the parameter is treated as not continuous but discrete. If a single number is input, then the parameter will not be optimised. The default is `seq(0.5, 1.5, length.out = 11)`.

`nmax` defines the maximum number of `data` observations to use when estimate each point using local kriging. It can be a scalar number, a vector of two values defining the optimisation range when the parameter is treated as a real number or a vector of length >2 values defining the optimisation increments when the parameter is treated as not continuous but discrete. If a single number is input, then the parameter will not be optimised. The default is `ceiling(seq(0.1,0.20,0.01)*length(data))`.

`nmax.fixedHead` defines the maximum number of `data.fixedHead` observations to use when estimate each point using local kriging. It can be a scalar number, a vector of two values defining the optimisation range when the parameter is treated as a real number or a vector of length >2 values defining the optimisation increments when the parameter is treated as not continuous but discrete. If a single number is input, then the parameter will not be optimised. The default is `seq(10,110,length=11)`.

`maxdist` defines the maximum search radius to use when estimate each point using local kriging. It can be a scalar number, a vector of two values defining the optimisation range when the parameter is treated as a real number or a vector of length >2 values defining the optimisation increments when the parameter is treated as not continuous but discrete. If a single number is input, then the parameter will not be optimised. The default is from 10% to 100% of `grid` extend at increments of 10%. If `grid` is `NULL`, then the user must input the search radius in one of the three accepted forms.

`trendMaxDistFrac`

defines a threshold distance between an observation and estimation point (as a fraction of `maxdist`) at which minimal weight should be placed on the observation (see Rivoirard et al. 2011). This can be used to reduce the numerical artefacts in the mapped head, which can appear as a linear sharp change in the head. It can be a scalar number, a vector of two values defining the optimisation range when the parameter is treated as a real number or a vector of length >2 values defining the optimisation increments when the parameter is treated as not continuous but discrete. If a single number is input, then the parameter will not be optimised. The default is `seq(0.1,1,0.1)`.

`nmin` defines the minimum number of `data` observations to use when estimating each point using local kriging. If the `nmin` observations cannot be located within the search radius `nmax`, then the search radius is increased until `nmin` points are obtained. If `nmin` is between zero and one and `omax` is between zero and one, then `nmin` is treated as a fraction of `nmax`. Else, `nmax` is treated as an integer number of data points and hence must be >0. This input cannot be optimised. The default value is 0.2.

`omax` defines the maximum number of `data` observations to select per quadrant when estimating each point using local kriging. It must be either `NULL` (to not use quadrant search) or between zero and one, which is treated as a fraction of `nmax`. This input cannot be optimised. The default value is `NULL`.

`nsim` defines the number of conditional simulations to undertake. If set to a non-zero value, conditional simulation is used instead of kriging interpolation. Importantly, this feature has not been tested. The default value is 0.

`fit.variogram.type` defines the way the model variogram is to be derived/used. For `fit.variogram.type==1` the input `model` must be the variogram type (as character string) and optimisation must be undertaken. The variogram will be assumed isotropic. For more control of the variogram calibrate using `krige.head.calib`. For `fit.variogram.type==2` the input `model` must be a `gstat` variogram model object of type `variogramModel`. The variogram model parameters will be estimated by fitting the model variogram to an experimental variogram using multi-start local calibration. For `fit.variogram.type==3` the input `model` must also be a variogram model object of type `variogramModel`. If calibration is being undertaken, then the variogram model parameters will not be optimised or fit to an experimental variogram.

`objFunc.type` defines the type of objective function to use in the optimisation. See `krige.head.calib` for details.

`use.cluster` sets if the calibration and interpolation should be parallelised. If `TRUE`, then local all local cores will be used. An integer >0 sets the number of local cores to use. An object from `makeCluster` allows for a user specified cluster.

`debug.level` Control the user messages. A value >0 outputs `hydroMap` progress. See `gstat` for the influence of values >0 on the kriging.

Details

This function is the primary means for using the package. It interpolates sparse input groundwater elevation observations for a single time point using a form of kriging with external drift that can account for land surface elevation, topographic form (e.g. valleys and ridges), the smoothness of the groundwater relative to the land surface and remote sensing gridded data. Also, a co-kriging features allows for the inclusion of categorical land types and fixed head boundary conditions, such as the ocean. Each of these features can be individually controlled by the user.

Importantly, if the mapping parameters are not specified by the user, then this function estimates the parameters using a mixed data-type (i.e. real and integer parameters) split-sample maximum likelihood global optimisation. The optimisation by default includes the variogram parameters (e.g. range, sill and nugget) and the search parameters for local kriging (e.g. radius, minimum and maximum number of observations to use). Optimising these parameters is not common in kriging. It is done herein because trials for Victoria, Australia, showed that calibrating these parameters produced significantly lower cross-validation errors (i.e. the error in predicting the observations removed from the optimisation) compared to the standard approach of graphical estimation from an experimental variogram. The optimisation is numerically challenging and the following factors should be considered before use:

- Optimisation of the parameters `mrvbf.pslope`, `mrvbf.ppctl` and `smooth.std` often required the creating of raster grids for every parameter combination. To ease the computation burden, these parameters should be treated as discrete, not continuous, numbers.
- The optimisation package `rgeoud` is used herein. For control the optimisation process, consider directly using `krige.head.calib`.

- Trials have established default calibration parameters and settings that were effective for Victoria, Australia. There is no guarantee they will be effective for other regions.

In using this function, the primary user decisions are:

- The kriging with external drift formula defining the independent gridded variables deemed to predict the groundwater elevation. See the input `formula`.
- The mapping extent and resolution, defined by the input `grid`, and the point observations of groundwater elevation, defined by the input `data`.
- The type of variogram model, defined by the input `model`

Value

If `is.null(newdata)==TRUE`, then a Spatial object grid will be returned with "head" and "head.var" for the groundwater level and kriging variance respectively.

Else, a point Spatial object will be returned with the estmates at the prediction locations and error estimates.

References

Gallant, J.C., Dowling, T.I. (2003): 'A multiresolution index of valley bottom flatness for mapping depositional areas', Water Resources Research, 39/12:1347-1359

Rivoirard, J. & Romary, T. Math Geosci (2011) Continuity for Kriging with Moving Neighborhood, Mathematical Geosciences, 43: 469. DOI: 10.1007/s11004-011-9330-0

See Also

`krige.head.calib` for undertaking only the optimisation.

Examples

```
# Load packages in case they have not loaded.
library(sp)
library(grid)
library(gstat)
library(raster)
library(RSAGA)
library(parallel)
library(rgenoud)

# Set enironment path for hydroMap
set.env()

# Load water table observations from  April 2000 for Victoria, Australia and a 250m state-wide DEM.
data('victoria.groundwater')

# Load a model variogram and mapping parametyers found to be effective.
data('mapping.parameters')
# Define a simple kriging formula without MrVBF terms that does not require the package RSAGA.
f <- as.formula('head ~ elev + smoothing')
```

```
# Interpolate the head data.
heads <- krige.head(formula=f, grid=DEM, data=obs.data, data.errvar.colname='total_err_var',
model=model, smooth.std=smooth.std, maxdist=maxdist, nmax=nmax, fit.variogram.type=3, debug.level=1)

# Recalibrate the parameters and map using the default settings.
heads <- krige.head(formula=f, grid=DEM, data=obs.data, data.errvar.colname='total_err_var',
model = 'Mat',  fit.variogram.type=1, debug.level=1)
```

---

krige.head.calib *Calibration of spatial interpolation parameters for mapping groundwater level observations.*

Description

krige.head.calib calibrates parameters to minimise the interpolation error.

Usage

```
krige.head.calib(formula = as.formula("head ~ elev + MrVBF + MrRTF + smoothing"),
grid, grid.landtype.colname = NULL, data, newdata = 0.5,
data.fixedHead = NULL, data.errvar.colname = NULL, model = c("Mat"),
mrvbf.pslope = if (any(match(all.vars(as.formula(formula)), "MrVBF", nomatch
= F) | match(all.vars(as.formula(formula)), "MrRTF", nomatch = F))) {
seq(0.5, 1.5, length.out = 11) } else {     NULL }, mrvbf.ppctl = if
(any(match(all.vars(as.formula(formula)), "MrVBF", nomatch = F) |
match(all.vars(as.formula(formula)), "MrRTF", nomatch = F))) {     seq(0.5,
1.5, length.out = 11) } else {     NULL }, smooth.std = seq(0.5, 1.5,
length.out = 11), nmax = if (is.character(data)) {     -999 } else {
ceiling(seq(0.1, 0.2, 0.01) * length(data)) }, nmax.fixedHead = if
(!is.null(data.fixedHead)) {     seq(10, 110, length = 11) } else {     NULL
}, maxdist = if (class(grid) == "SpatialPixelsDataFrame" || class(grid) ==
"SpatialGridDataFrame") {     ceiling(0.5 * sqrt((extent(grid)[2] -
extent(grid)[1])^2 + (extent(grid)[4] - extent(grid)[3])^2) * seq(0.1, 1,
0.1)) } else {     -999 }, trendMaxDistFrac = seq(0.1, 1, 0.1),
nmin = 0.2, omax = NULL, fit.variogram.type = 1, objFunc.type = 1,
use.cluster = TRUE, pop.size.multiplier = 3, max.generations = 100,
hard.generation.limit = FALSE, solution.tolerance = 1e-04,
debug.level = 0)
```

Arguments

formula	see krige.head.
MrVBF	see krige.head.
MrRTF	see krige.head.

`smoothing` see krige.head.

`grid` see krige.head.

`grid.landtype.colname`
see krige.head.

`data` see krige.head.#'

`data.fixedHead` see krige.head.

`newdata` is as for `data` but the points in a split-sample cross-validation scheme to estimate the interpolation error. Points listed within `newdata` should not be listed within `data`. `newdata` can be (i) a real scalar >0 and <1 defining the fraction of data points within `data` to randomly remove and from `data` and use for cross-validation, (i) an integer scalar >1 and <`length(data)` defining the number of data points within `data` to randomly remove and from `data` and use for cross-validation (iii) a vector of indexes defining rows numbers within `data` to be extract and uses for the cross-validation; (iv) a vector of logicals with \ciode-TRUE defining rows numbers within `data` to be extract and used for the cross-validation, (v) a `SpatialPointsDataFrame` variable defining the complete data for cross validation - which must have identical columns to `data`. The default is `newdata=0.5`,

`data.errvar.colname`
see krige.head.

`model` is either a character for the name of the variogram model type or a `gstat` variogram model object of type `variogramModel`. The available options are as per `gstat`, but it is suggested to be `Mat`.

`mrvbf.pslope` defines the calibration type and range for the MrVFB shape parameter for the slope (see Gallant et al. 2003). A vector of two values defining the optimisation range when the parameter is treated as a real number or a vector of length >2 values defining the optimisation increments when the parameter is treated as not continuous but discrete. If a single number is input, then the parameter will not be optimised. If the `formula` includes either of the terms `MrVBF` or `MrRTF`, then the default is `seq(0.5, 1.5, length.out = 11)`. Else, the default is `NULL`.

`mrvbf.ppctl` defines the calibration type and range for the MrVFB shape parameter for elevation percentile (see Gallant et al. 2003). It can be a scalar number, a vector of two values defining the optimisation range when the parameter is treated as a real number or a vector of length >2 values defining the optimisation increments when the parameter is treated as not continuous but discrete. If a single number is input, then the parameter will not be optimised. If the `formula` includes either of the terms `MrVBF` or `MrRTF`, then the default is `seq(0.5, 1.5, length.out = 11)`. Else, the default is `NULL`.

`smooth.std` defines the calibration type and range for the strength of the Gaussian kernal smoothing applied to the 5x5 grid cells surrounding each DEM grid cell. It can be a scalar number, a vector of two values defining the optimisation range when the parameter is treated as a real number or a vector of length >2 values defining the optimisation increments when the parameter is treated as not continuous but discrete. If a single number is input, then the parameter will not be optimised. The default is `seq(0.5, 1.5, length.out = 11)`.

nmax defines the calibration type and range for the maximum number of data observations to use when estimate each point using local kriging. It can be a scalar number, a vector of two values defining the optimisation range when the parameter is treated as a real number or a vector of length >2 values defining the optimisation increments when the parameter is treated as not continuous but discrete. If a single number is input, then the parameter will not be optimised. The default is ceiling(seq(0.1,0.20,0.01)*length(data)).

nmax.fixedHead defines the calibration type and range for the maximum number of data.fixedHead observations to use when estimate each point using local kriging. It can be a scalar number, a vector of two values defining the optimisation range when the parameter is treated as a real number or a vector of length >2 values defining the optimisation increments when the parameter is treated as not continuous but discrete. If a single number is input, then the parameter will not be optimised. The default is seq(10,110,length=11).

maxdist defines the calibration type and range for the maximum search radius to use when estimate each point using local kriging. It can be a scalar number, a vector of two values defining the optimisation range when the parameter is treated as a real number or a vector of length >2 values defining the optimisation increments when the parameter is treated as not continuous but discrete. If a single number is input, then the parameter will not be optimised. The default is from 10% to 100% of grid extend at increments of 10%. If grid is NULL, then the user must input the search radius in one of the three accepted forms.

trendMaxDistFrac
defines the calibration type and range for the threshold parameter for the distance between an observation and estimation point (as a fraction of maxdist) at which minimal weight should be placed on the observation (see Rivoirard et al. 2011). This can be used to reduce the numerical artefacts in the mapped head, which can appear as a linear sharp change in the head. It can be a scalar number, a vector of two values defining the optimisation range when the parameter is treated as a real number or a vector of length >2 values defining the optimisation increments when the parameter is treated as not continuous but discrete. If a single number is input, then the parameter will not be optimised. The default is seq(0.1,1,0.1).

nmin see krige.head.

omax see krige.head.

nsim see krige.head.

fit.variogram.type
defines the way the model variogram is to be derived/used. For fit.variogram.type==1 the input model then all of the variogram parameters will be calibrated. When model is a chacater for a variogram modle type, then the variogram will be assumed isotropic. When model is a gstat variogram model object of type variogramModel, then all of the variogram parameters will be calibrated. This approach allows significantly greater control of the variogram model form. For fit.variogram.type==2 the input model must be a gstat variogram model object of type variogramModel. The variogram model parameters will not be calibrated and instead will be estimated by fitting the model variogram to an experimental variogram using multi-start local calibration. This approach is very

similar to convential variogram model fitting. For `fit.variogram.type==3` the input `model` must also be a variogram model object of type `variogramModel`. The variogram model parameters will not be calibrated or fit to an experimental variogram. The default is `fit.variogram.type=1`.

`objFunc.type` defines the type of objective function to use in the calibration. For `objFunc.type==1`, the negative likelihood divided by the number of `newdata` observations is used - which accounts for the expected error in the `newdata` estimates. That is, `newdata` points far from other `data` points will have a large kriging variance and hence the expected error in the kriging estimate is larger. Conversely, `newdata` points near to `data` points will have a low kriging variance and hence the kriging water level error should be low. For details see Samper and Neumen (1989A,B) and Pardo-Iguzquiza & Dowd (2013). For `objFunc.type==2`, it is as for type 1 but with an added penalty when the estimate violates a physical constraint. by first estimating the difference between the land surface elevation and the head. At those `newdata` points above the land surface, the absolute of this difference is added to the absolute error between the predicated value and that within `newdata`. For `objFunc.type==3`, the root-mean-square error is used. For `objFunc.type==4`, the root-mean-square error is used but with the penalty from type 2. The default is `objFunc.type=1`.

`use.cluster` see `krige.head`.

`debug.level` see `krige.head`.

Value

As per genoud plus the parameter values in the transformed calibration space "par.pretransformed" and "par" is replaced as a named list variable.

References

Gallant, J.C., Dowling, T.I. (2003) A multiresolution index of valley bottom flatness for mapping depositional areas, Water Resources Research, 39/12:1347-1359

Rivoirard, J. & Romary, T. Math Geosci (2011) Continuity for Kriging with Moving Neighborhood, Mathematical Geosciences, 43: 469. DOI: 10.1007/s11004-011-9330-0

Pardo-Igúzquiza E. , Dowd P. A., (2013) Comparison of inference methods for estimating semi-variogram model parameters and their uncertainty: The case of small data sets, Computers & Geosciences, v50,pp 154-164, https://doi.org/10.1016/j.cageo.2012.06.002.

Samper, F. J., and S. P. Neuman (1989A), estimation of spatial covariance structures by adjoint state maximum likelihood cross validation: 1. Theory, Water Resour. Res., 25(3), 351–362, doi: 10.1029/WR025i003p00351.

Samper, F. J., and S. P. Neuman (1989B), Estimation of spatial covariance structures by adjoint state maximum likelihood cross validation: 2. Synthetic experiments, Water Resour. Res., 25(3), 363–371, doi: 10.1029/WR025i003p00363.

See Also

`krige.head` for undertaking the mapping.

Examples

```
# Load packages in case they have not loaded.
library(sp)
library(grid)
library(gstat)
library(raster)
library(RSAGA)
library(parallel)
library(rgenoud)

# Set enironment path for hydroMap
set.env()

# Load water table observations from  April 2000 for Victoria, Australia and a 250m state-wide DEM.
data('victoria.groundwater')

# Load a model variogram and mapping parameters found to be effective.
data('mapping.parameters')
# Define a simple kriging formula without MrVBF terms that does not require the package RSAGA.
f <- as.formula('head ~ elev + smoothing')

# Define an initial isotropic variogram model. All of the parameters will be calibrated.
varigram.model <- vgm(psill=25, model="Mat", range=5000, nugget=5, kappa = 0.5)

# Calibrate the mapping parameters with 25% of the data randomly selected.
calib.results <- krige.head.calib(formula=f, grid=DEM, data=obs.data, newdata=0.25,
data.errvar.colname='total_err_var', model = varigram.model,  fit.variogram.type=1,
debug.level=1)

# Reformat the parameter to a named list. This is just for ease of use and the need for
this may be removed in future versions.
params.all = c(mrvbf.pslope=NA,mrvbf.ppctl=NA,smooth.std=NA,nmax=NA,maxdist=NA,
               trendMaxDistFrac=NA,nug = NA, psill_model_1 = NA, range_model_1 = NA,
               kappa_model_1 = NA, ang1_model_1 = NA, anis1_model_1 = NA,
               nmax.fixedHead = NA);
for (i in 1:length(head.calib$param.Names)) {
 params.all[[calib.results$param.Names[i]]] = calib.results$par[i]
}
params.all.names=names(params.all);

# Rebuild the variogram with the calibrated parameters.
varigram.model = vgm(psill=params.all[['psill_model_1']], model='Mat',
                  range= params.all[['range_model_1']] , nugget=params.all[['nug']],
                  kappa=params.all[['kappa_model_1']] );

# Grid the observed head data.
head.map <- krige.head(formula=f, grid=DEM, data=obs.data, data.errvar.colname='total_err_var',
model=varigram.model, smooth.std=params.all[['smooth.std']], maxdist=params.all[['maxdist']],
nmax=params.all[['nmax']], fit.variogram.type=3, debug.level=1)

# Write the grids to ARCMAP ASCII grids
write.asciigrid(head.map, 'head.asc',1);
```

```
write.asciigrid(head.map, 'KrigingVariance.asc',2);
```

Index