Package ‘gcKrig’

December 21, 2018

Type Package

Title Analysis of Geostatistical Count Data using Gaussian Copulas

Version 1.1.3

Author Zifei Han

Maintainer Zifei Han <hanzifei@gmail.com>

Description Provides a variety of functions to analyze and model geostatistical count data with Gaussian copulas, including
1) data simulation and visualization;
2) correlation structure assessment (here also known as the Normal To Anything);
3) calculate multivariate normal rectangle probabilities;
4) likelihood inference and parallel prediction at predictive locations.

License GPL (>= 2)

LazyData TRUE

Depends R (>= 3.2.5)

Imports Rcpp (>= 0.12.0)

LinkingTo Rcpp, RcppArmadillo

Suggests EQL, FNN, lattice, latticeExtra, Matrix, MASS, numDeriv, scatterplot3d, snowfall, sp

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-12-21 12:00:03 UTC

R topics documented:

AtlanticFish ......................................................... 2
beta.gc .......................................................... 3
binomial.gc ....................................................... 4
corr.gc ........................................................... 5
corrTG ............................................................. 6
FHUBdiscrete ....................................................... 7
gaussian.gc ....................................................... 8
Description

This dataset was studied by Johnson and Hoeting (2011) for analyzing pollution tolerance in Mid-Atlantic Highlands Fish. Pollution intolerant fish were sampled, and several stream characteristics were measured to assess water quality at 119 sites in the Mid-Atlantic region of the United States. All covariates of the data had been standardized to have mean 0 and variance 1.

Usage

data(AtlanticFish)

Format

A data frame with 119 observations and 12 variables.

LON Longitude of the location.
LAT Latitude of the location.
ABUND Fish abundance at given locations, the discrete response.
ORDER Strahler stream order, a natural covariate measuring stream size.
DISTOT Watershed classified as disturbed by human activity, a variable reflecting stream quality.
HAB An index of fish habitat quality at the stream site, a variable reflecting stream quality.
WSA Watershed area, a natural covariate.
ELEV Elevation.
RD Road density in the watershed, a variable reflecting stream quality.
DO Concentration of dissolved oxygen in the stream at the sampling site, a stream quality variable.
XFC Percent of areal fish cover at the sampling site, a stream quality variable.
PCT Percent of sand in streambed substrate, a stream quality variable.

References

Examples

```
data(AtlanticFish)
str(AtlanticFish)
```

---

**beta.gc**  
*The Beta Marginal of Class marginal.gc*

**Description**

The Beta marginal used for simulation and computing correlation in the trans-Gaussian random field in function simgc and corrT6 of the package gcKrig. It cannot be used in function mlegc nor predgc to make model inferences.

**Usage**

```
beta.gc(shape1 = 1, shape2 = 1)
```

**Arguments**

- `shape1`: non-negative scalar, the shape parameter of the Beta distribution.
- `shape2`: non-negative scalar, another shape parameter of the Beta distribution.

**Details**

The Beta distribution with parameters `shape1 = a` and `shape2 = b` has density

\[
\frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1 - x)^{b-1}
\]

for \( a > 0, b > 0 \) and \( 0 \leq x \leq 1 \) where the boundary values at \( x = 0 \) or \( x = 1 \) are defined as by continuity (as limits).
Description
The binomial marginal parameterized in terms of its size and probability.
By default, this function is used for likelihood inference and spatial prediction in function mlegc and predgc of the package gckrig. When all marginal parameters are given, the function is used for simulation and computing correlation in a trans-Gaussian random field in function simgc and corrTG.

Usage
binomial.gc(link = "logit", size = NULL, prob = NULL)

Arguments
link the model link function.
size number of trials (zero or more).
prob probability of success on each trial.

Value
An object of class marginal.gc representing the marginal component.

Author(s)
Zifei Han <hanzifei1@gmail.com>

See Also
marginal.gc, binomial.gc, gm.gc, gaussian.gc, negbin.gc, poisson.gc, weibull.gc, zip.gc
Description

Class of isotropic correlation functions available in the gcKrig library.

Details

By default, range parameter is not provided, so this function is used for likelihood inference and spatial prediction with function mlegc and predgc. Users need to specify if the correlation model includes a nugget effect nugget = TRUE or not nugget = FALSE. For Matern and powered exponential correlation functions, the shape parameter kappa is also required from users.

When both range and nugget parameters are given, the function is used to specify the correlation structure in simulation with function simgc in package gcKrig.

Value

At the moment, the following three correlation functions are implemented:

- matern.gc: the Matern correlation function.
- powerexp.gc: the powered exponential correlation function.
- spherical.gc: the spherical correlation function.

Author(s)

Zifei Han <hanzifei@gmail.com>

References


See Also

matern.gc, powerexp.gc, spherical.gc
**corrTG**

*Compute the Correlation in Transformed Gaussian Random Fields*

**Description**

This function implements two general methods for computing the correlation function in a transformed Gaussian random field.

**Usage**

```r
corrTG(marg1, marg2, corrGauss = 0.5, method = "integral", nrep = 1000)
```

**Arguments**

- `marg1`: an object of class `marginal.gc` specifying the first marginal distribution.
- `marg2`: an object of class `marginal.gc` specifying the second marginal distribution.
- `corrGauss`: the correlation in the Gaussian random field. Should be a scalar between 0 and 1.
- `method`: the computation method of calculating correlation in the transformed Gaussian random field. Can be either "integral" or "mc". If use "integral" then a series expansion based on the Hermite Polynomials will be used to approximate the correlation, see De Oliveira (2013). If use "mc" then the Monte Carlo method will be used.
- `nrep`: the Monte Carlo size in computing the correlation. Only need to be specified if `method` = "mc".

**Value**

A scalar between 0 and 1, denoting the correlation of the transformed Gaussian random field.

**Author(s)**

Zifei Han <hanzifei@gmail.com>

**References**


Examples

```r
## Not run:
corrTG(marg1 = poisson.gc(lambda = 10), marg2 = binomial.gc(size = 1, prob = 0.1),
corrGauss = 0.5, method = "integral")
set.seed(12345)
corrTG(marg1 = poisson.gc(lambda = 10), marg2 = binomial.gc(size = 1, prob = 0.1),
corrGauss = 0.5, nrep = 100000, method = "mc")
## End(Not run)
```

### FHUBdiscrete

**Compute the Frechet Hoeffding Upper Bound for Given Discrete Marginal Distributions**

**Description**

This function implemented the method of computing the Frechet Hoeffding upper bound for discrete marginals described in Nelsen (1987), which can only be applied to discrete marginals. Four commonly used marginal distributions were implemented. The distribution "nb" (negative binomial) and "zip" (zero-inflated Poisson) are parameterized in terms of the mean and overdispersion, see Han and De Oliveira (2016).

**Usage**

```r
FHUBdiscrete(marg1, marg2, mu1, mu2, od1 = 0, od2 = 0, binomial.size1 = 1,
             binomial.size2 = 1)
```

**Arguments**

- `marg1`: name of the first discrete marginal distribution. Should be one of the "poisson", "zip", "nb" or "binomial".
- `marg2`: name of the second discrete marginal distribution. Should be one of the "poisson", "zip", "nb" or "binomial".
- `mu1`: mean of the first marginal distribution. If binomial then it is \( n_1 p_1 \).
- `mu2`: mean of the second marginal distribution. If binomial then it is \( n_2 p_2 \).
- `od1`: the overdispersion parameter of the first marginal. Only used when marginal distribution is either "zip" or "nb".
- `od2`: the overdispersion parameter of the second marginal. Only used when marginal distribution is either "zip" or "nb".
- `binomial.size1`: the size parameter (number of trials) when marg1 = "binomial".
- `binomial.size2`: the size parameter (number of trials) when marg2 = "binomial".

**Value**

A scalar denoting the Frechet Hoeffding upper bound of the two specified marginal.
The Gaussian Marginal of Class marginal.gc

description

The Gaussian marginal used for simulation and computing correlation in the trans-Gaussian random field in function `simgc` and `corrtG` of the package `gcKrig`. It cannot be used in function `mlegc` nor `predgc` to make model inferences.

Usage

```r
gaussian.gc(mean = 0, sd = 1)
```

Arguments

- `mean`: the mean of the Gaussian distribution, a scalar.
- `sd`: a positive scalar, the standard deviation of the Gaussian distribution.
gm.gc

Value
An object of class marginal.gc representing the marginal component.

Author(s)
Zifei Han <hanzifei1@gmail.com>

See Also
marginal.gc, beta.gc, binomial.gc, gm.gc, negbin.gc, poisson.gc, weibull.gc, zip.gc

Description
The Gamma marginal used for simulation and computing correlation in the trans-Gaussian random field in functions simgc and corrtG of the package gckrig. It cannot be used in functions mlegc nor predgc to make model inferences.

Usage
gm.gc(shape = 1, rate = 1)

Arguments
shape a non-negative scalar, shape parameter of the Gamma distribution.
rate a non-negative scalar, rate parameter of the Gamma distribution.

Details
The Gamma distribution with parameters shape = a and rate = r has density

\[
\frac{r^a}{\Gamma(a)} x^{a-1} \exp(-rx)
\]

for \(x \geq 0, a > 0\) and \(s > 0\).

Value
An object of class marginal.gc representing the marginal component.

Author(s)
Zifei Han <hanzifei1@gmail.com>

See Also
marginal.gc, beta.gc, binomial.gc, gaussian.gc, negbin.gc, poisson.gc, weibull.gc, zip.gc
Description

The data is aggregated from the dataset `lansing` in library `spatstat`, which came from an investigation of a 924 ft x 924 ft (19.6 acres) plot in Lansing Woods, Clinton County, Michigan USA by D.J. Gerrard. The original point process data described the locations of 2,251 trees and their botanical classification (into maples, hickories, black oaks, red oaks, white oaks and miscellaneous trees). The original plot size has been rescaled to the unit square and the number of different types of trees has been counted within squares of length 1/16.

Usage

data(LansingTrees)

Format

A data frame with 256 observations and 8 variables.

- `easting` Cartesian x-coordinate of the locations.
- `northing` Cartesian y-coordinate of the locations.
- `maple` Number of maples in the area.
- `hickory` Number of hickories in the area.
- `blackoak` Number of black oaks in the area.
- `redoak` Number of red oaks in the area.
- `whiteoak` Number of white oaks in the area.
- `misc` Number of miscellaneous trees in the area.

References


Examples

data(LansingTrees)
str(LansingTrees)
Class of marginals available in gcKrig library for geostatistical data simulation, correlation structure assessment (both continuous and discrete marginals) and model inferences (discrete marginals only). In former cases parameters of the marginals are given by users, otherwise parameters are estimated from the data (except when doing prediction with function predgc, users can choose to either input known estimates or estimate the parameters with input data).

By default, when the marginals are discrete, they are used for estimation with function mlegc and prediction with function predgc. They can be used in function simgc and corrtG as well for the purpose of data simulation and correlation computation in a transformed Gaussian random field (Han and De Oliveira, 2016), if parameter values are specified.

For continuous marginals, they are used for simulation with function simgc and correlation computation with corrtG only, so parameters should always be specified.

At the moment, the following marginals are implemented:

- beta.gc: beta marginals.
- binomial.gc: binomial marginals.
- gm.gc: gamma marginals.
- gaussian.gc: Gaussian marginals.
- negbin.gc: negative binomial marginals.
- poisson.gc: Poisson marginals.
- weibull.gc: Weibull marginals.
- zip.gc: zero-inflated Poisson marginals.

Zifei Han <hanzifei@gmail.com>


See Also

`betaNgc`, `binomialNgc`, `gmNgc`, `gaussianNgc`, `negbinNgc`, `poissonNgc`, `weibullNgc`, `zipNgc`

---

**maternNgc**

*The Matern Correlation Function of Class corrNgc*

**Description**

The Matern correlation function in spatial statistics.

By default, range parameter is not available, so this function is used for likelihood inference and spatial prediction in function `mlegc` and `predgc`. Users need to specify the shape parameter `kappa` and if the correlation model includes a nugget effect `nugget = TRUE` or not `nugget = FALSE`.

When both range and nugget parameters are given, the function is used for simulation with function `simgc` in package `gcKrig`.

**Usage**

```r
maternNgc(range = NULL, kappa = 0.5, nugget = TRUE)
```

**Arguments**

- `range` a non-negative scalar of the range parameter in Matern correlation function.
- `kappa` a non-negative scalar of the shape parameter in the Matern correlation function. The default `kappa = 0.5` corresponds to an exponential correlation model.
- `nugget` the nugget effect of the correlation function. If specified, it must be a scalar between 0 and 1.
mlegc

Details

The Matern correlation function with a nugget $\tau^2$ is of the form:

$$
\rho(h) = 1 - \frac{\tau^2}{2^{\kappa-1}\Gamma(\kappa)} \left( \frac{h}{\phi} \right)^\kappa K_\kappa \left( \frac{h}{\phi} \right)
$$

when $h > 0$ and $\rho(h) = 1$ when $h = 0$. Here $\phi$ is range parameter, $\kappa$ is the shape parameter and $\tau^2$ is the nugget parameter. $K_\kappa(\cdot)$ denotes the modified Bessel function of the third kind of order $\kappa$.

Value

An object of class \texttt{corr.gc} representing the correlation component.

Author(s)

Zifei Han <hanzifei@gmail.com>

References


See Also

\texttt{powerexp.gc, spherical.gc}

---

\texttt{mlegc} \quad \textit{Maximum Likelihood Estimation in Gaussian Copula Models for Geostatistical Count Data}

Description

Computes the maximum likelihood estimates. Two methods are implemented. If method = 'GHK' then the maximum simulated likelihood estimates are computed, if method = 'GQT' then the maximum surrogate likelihood estimates are computed.

Usage

\texttt{mlegc(y, x = NULL, locs, marginal, corr, effort = 1, longlat = FALSE, distscale = 1, method = "GHK", corrpar0 = NULL, ghkoptions = list(nrep = c(100, 1000), reorder = FALSE, seed = 12345))}
Arguments

- **y**: a non-negative integer vector of response with its length equals to the number of sampling locations.
- **x**: a numeric matrix or data frame of covariates, with its number of rows equals to the number of sampling locations. If no covariates then $x = \text{NULL}$.
- **locs**: a numeric matrix or data frame of $n$-D points with row denoting points. The first column is $x$ or longitude, the second column is $y$ or latitude. The number of locations is equal to the number of rows.
- **marginal**: an object of class `marginal.gc` specifying the marginal distribution.
- **corr**: an object of class `corr.gc` specifying the correlation function.
- **effort**: the sampling effort. For binomial marginal it is the size parameter (number of trials). See details.
- **longlat**: if FALSE, use Euclidean distance, if TRUE use great circle distance. The default is FALSE.
- **distscale**: a numeric scaling factor for computing distance. If original distance is in kilometers, then $\text{distscale} = 1000$ will convert it to meters.
- **method**: two methods are implemented. If method = 'GHK' then the maximum simulated likelihood estimates are computed, if method = 'GQT' then the maximum surrogate likelihood estimates are computed.
- **corrpar0**: the starting value of correlation parameter in the optimization procedure. If corrpar0 = NULL then initial range is set to be half of the median distance in distance matrix and initial nugget (if nugget = TRUE) is 0.2.
- **ghkoptions**: a list of three elements that only need to be specified if method = 'GHK'.

Details

This program implemented one simulated likelihood method via sequential importance sampling (see Masarotto and Varin 2012), which is same as the method implemented in package gcmr (Masarotto and Varin 2016) except an antithetic variable is used. It also implemented one surrogate likelihood method via distributional transform (see Kazianka and Pilz 2010), which is generally faster.

The argument effort is the sampling effort (known). It can be used to consider the heterogeneity of the measurement time or area at different locations. The default is 1 for all locations. See Han and De Oliveira (2016) for more details.
Value

A list of class "mlegc" with the following elements:

- **MLE** the maximum likelihood estimate.
- **x** the design matrix.
- **nug** 1 if nugget = TRUE, 0 if nugget = FALSE.
- **nreg** number of regression parameters.
- **log.lik** the value of the maximum log-likelihood.
- **AIC** the Akaike information criterion.
- **AICc** the AICc information criterion; essentially AIC with a greater penalty for extra parameters.
- **BIC** the Bayesian information criterion.
- **kmarg** number of marginal parameters.
- **par.df** number of parameters.
- **N** number of observations.
- **D** the distance matrix.
- **optlb** lower bound in optimization.
- **optub** upper bound in optimization.
- **hessian** the hessian matrix evaluated at the final estimates.
- **args** arguments passed in function evaluation.

Author(s)

Zifei Han <hanzifei1@gmail.com>

References


See Also

- gcnr
Examples

```r
## Not run:
## Fit a Simulated Dataset with 100 locations
grid <- seq(0.05, 0.95, by = 0.1)
xloc <- expand.grid(x = grid, y = grid)[,1]
yloc <- expand.grid(x = grid, y = grid)[,2]

set.seed(123)
simData1 <- simgc(locs = cbind(xloc,yloc), sim.n = 1,
marginal = negbin.gc(mu = exp(1+xloc), od = 1),
corr = matern.gc(range = 0.4, kappa = 0.5, nugget = 0))

simFit1 <- mlegc(y = simData1$data, x = xloc, locs = cbind(xloc,yloc),
marginal = negbin.gc(link = 'log'),
corr = matern.gc(kappa = 0.5, nugget = FALSE), method = 'GHK')

simFit2 <- mlegc(y = simData1$data, x = xloc, locs = cbind(xloc,yloc),
marginal = negbin.gc(link = 'log'),
corr = matern.gc(kappa = 0.5, nugget = FALSE), method = 'GQT')
#summary(simFit1);summary(simFit2)
#plot(simFit1);plot(simFit2)

## Time consuming examples
## Fit a real dataset with 70 sampling locations.
data(Weed95)
weedobs <- Weed95[Weed95$dummy==1, ]
weedpred <- Weed95[Weed95$dummy==0, ]
Weedfit1 <- mlegc(y = weedobs$weedcount, x = weedobs[,4:5], locs = weedobs[,1:2],
marginal = poisson.gc(link='log'),
corr = matern.gc(kappa = 0.5, nugget = TRUE),
method = 'GHK')

summary(Weedfit1)
plot(Weedfit1)

## Fit a real dataset with 256 locations
data(Lansingtrees)
Treefit1 <- mlegc(y = LansingTrees[,3], x = LansingTrees[,4], locs = LansingTrees[,1:2],
marginal = negbin.gc(link = 'log'),
corr = matern.gc(kappa = 0.5, nugget = FALSE), method = 'GHK')

summary(Treefit1)
plot(Treefit1)

# Try to use GQT method
Treefit2 <- mlegc(y = LansingTrees[,3], x = LansingTrees[,4],
locs = LansingTrees[,1:2], marginal = poisson.gc(link='log'),
corr = matern.gc(kappa = 0.5, nugget = TRUE), method = 'GQT')

summary(Treefit2)
plot(Treefit2)
```
## mvnintGHK

**Computing Multivariate Normal Rectangle Probability**

### Description

Computes the multivariate normal rectangle probability for arbitrary limits and covariance matrices using (reordered) sequential importance sampling.

### Usage

```r
mvnintGHK(mean, sigma, lower, upper, nrep = 5000, log = TRUE, reorder = TRUE)
```
Arguments

- **mean**: the numeric vector of mean of length n.
- **sigma**: the covariance matrix of dimension n.
- **lower**: the numeric vector of lower limits of length n.
- **upper**: the numeric vector of upper limits of length n.
- **nrep**: a positive integer of Monte Carlo size.
- **log**: if TRUE then return the log of the probability. If FALSE return the probability.
- **reorder**: if TRUE then variable reordering algorithm is applied. If FALSE then original ordering is used.

Details

This program implemented the Geweke-Hajivassiliou-Keane simulator of computing the multivariate normal rectangle probability. For more details see Keane (1994). Also a variable reordering algorithm in Gibson, etal (1994) was implemented.

Note that both -Inf and Inf may be specified in lower and upper.

Value

A list of the following two components:

- **value**: the value of the integral. If log = TRUE then output the log of the integral.
- **error**: the Monte Carlo standard deviation.

Author(s)

- Zifei Han <hanzifei@gmail.com>

References


See Also

- pmvnorm

Examples

```r
mvnintGHK(mean = rep(0, 51), sigma = diag(0.2, 51) + matrix(0.8, 51, 51), lower = rep(-2,51), upper = rep(2,51), nrep = 10000)
```
**Description**

The negative binomial marginal parameterized in terms of its mean and overdispersion.

By default, this function is used for likelihood inference and spatial prediction in functions `mlegc` and `predgc` of the package `gckrig`. When all marginal parameters are given, the function is used for simulation and computing correlation in a trans-Gaussian random field in functions `simgc` and `corrTG`.

**Usage**

```r
negbinNgc(link = "log", mu = NULL, od = NULL)
```

**Arguments**

- `link` the model link function.
- `mu` a non-negative scalar of the mean parameter.
- `od` a non-negative scalar of the overdispersion parameter.

**Details**

The negative binomial distribution with parameters \( \mu = a \) and \( \phi = 1/b \) has density

\[
\frac{\Gamma(y + b)}{\Gamma(b) y!} \left(\frac{b}{b + a}\right)^b \left(1 - \frac{b}{b + a}\right)^y
\]

which is called NB2 by Cameron and Trivedi (2013). Under this parameterization, \( \text{var}(Y) = \mu + \phi \mu^2 \), where \( \mu \) is the mean parameter and \( \phi \) is the overdispersion parameter. For more details see Han and De Oliveira (2016).

**Value**

An object of class `marginal.gc` representing the marginal component.

**Author(s)**

Zifei Han <hanzifei1@gmail.com>

**References**


See Also

marginal.gc, beta.gc, binomial.gc, gm.gc, gaussian.gc, poisson.gc, weibull.gc, zip.gc

OilWell

Location of Successful and Dry Wells

Description

A dataset recording locations of successful and unsuccessful drilling oil wells in the northwest shelf of Delaware basin in New Mexico, a region that is densely drilled but has some sparsely drilled areas. The original dataset was transformed to a central area of about 65 square kilometers, see Hohn (1999), Chapter 6.

Usage

data(OilWell)

Format

A data frame with 333 observations and 3 variables.

Easting  Cartesian x-coordinate of the locations.

Northing  Cartesian y-coordinate of the locations.

Success  A binary variable indicating the success of the drill at given locations. 1 for successful drilling oil wells and 0 for unsuccess.

References


Examples

data(OilWell)

str(OilWell)
Description

Four plots can be generated: the 2-D level plot or 3-D scatterplot with the number of counts or fitted values.

Usage

```r
## S3 method for class 'mlegc'
plot(x, plotdata = "Observed", plottype = "2D", xlab = "xloc", ylab = "yloc",
    xlim = NULL, ylim = NULL, pch = 20, textcex = 0.8, plotcex = 1,
    angle = 60, col = 4, col.regions = gray(90:0/100),...)
```

Arguments

- `x`: an object of class `mlegc` inherited from function `mlegc`.
- `plotdata`: the data to be plotted. Can be either "Observed" if the original counts are used or "Fitted" if the fitted mean at different locations are used.
- `plottype`: the type of the plot. Can be either "2D" for 2-D level plot or "3D" for 3-D scatterplot.
- `xlab`, `ylab`: a title for the x and y axis.
- `xlim`, `ylim`: numeric vectors of length 2, giving the x and y coordinates ranges. if they equal to NULL then they will be adjusted from the data.
- `pch`: plotting character, i.e., the symbol to use in the 3-D scatter plot.
- `textcex`: a numerical value giving the amount by which plotting text should be magnified relative to the default.
- `plotcex`: a numerical value giving the amount by which plotting symbols should be magnified relative to the default.
- `angle`: angle between x and y axis.
- `col`: color of the text.
- `col.regions`: color vector to be used reflecting magnitude of the dataset at different locations. The general idea is that this should be a color vector of gradually varying color.
- `...`: further arguments passed to plot and panel settings.

Author(s)

Zifei Han <hanzifei@gmail.com>

See Also

`plot.mimgc, plot.predgc`
Description

Five plots can be generated. A level plot with the number of counts at both observed and prediction locations; a level plot with predicted means (intensity); a level plot with the predicted counts; a level plot with estimated variances of the prediction; a 3-D scatter plot with both observed and predicted counts.

Usage

```r
## S3 method for class 'predgc'
plot(x, plottype = "2D", xlab = "xloc", ylab = "yloc", xlim = NULL,
ylim = NULL, pch = 20, textcex = 0.6, plotcex = 1, angle = 60,
col = c(2, 4), col.regions = gray(90:0/100),...)
```

Arguments

- `x`: an object of class `predgc` inherited from function `predgc`.
- `plottype`: can be one of the following: "2D", "Predicted Counts", "Predicted Mean", "Predicted Variance" or "3D". Default is "2D" which generates a 2-D contour plot with both observed and predicted counts. With arguments "Predicted Counts", "Predicted Mean" and "Predicted Variance", a 2-D level plot will be generated with the corresponding data. When "3D" is used, a 3-D scatter plot will be displayed with observed and predicted counts.
- `xlab`, `ylab`: a title for the x and y axis.
- `xlim`, `ylim`: numeric vectors of length 2, giving the x and y coordinates ranges. if they equal to `NULL` then they will be adjusted from the data.
- `pch`: plotting character, i.e., symbol to use in the 3-D scatter plot.
- `textcex`: a numerical value giving the amount by which plotting text should be magnified relative to the default.
- `plotcex`: a numerical value giving the amount by which plotting symbols should be magnified relative to the default.
- `angle`: angle between x and y axis.
- `col`: a numeric vector of length 2 indicating color of the plot at sampling and prediction locations.
- `col.regions`: color vector to be used reflecting magnitude of the dataset at different locations. The general idea is that this should be a color vector of gradually varying color.
- `...`: further arguments passed to plot and panel settings.

Author(s)

Zifei Han <hanzifei@gmail.com>
See Also

plot.simgc, plot.mlegc, mlegc, predgc

---

plot.simgc  

**Plot Geostatistical Data Simulated From Gaussian Copula**

---

**Description**

Three plots will be generated. A level plot with the number of counts at given locations; a level plot with point referenced locations and varying colors and a 3-D scatter plot.

**Usage**

```r
## S3 method for class 'simgc'
plot(x, index = 1, plottype = "Text", xlab = "xloc", ylab = "yloc",
exlim = NULL, ylim = NULL, pch = 20, textcex = 0.8, plotcex = 1,
angle = 60, col = 4, col.regions = gray(90:0/100),...)
```

**Arguments**

- `x` an object of class `simgc`, typically generated from function `simgc`.
- `index` the index of the simulated data, need to be specified since `simgc` can simulate multiple datasets simultaneously.
- `plottype` the type of the printed plot, can be "Text", "Dot", or "3D". When `plottype = "Text"`, a 2-D plot is generated with exact counts at observed locations. When `plottype = "Dot"`, a 2-D dot plot is generated and when `plottype = "3D"` a 3-D scatter plot is printed.
- `xlab`, `ylab` a title for the x and y axis.
- `xlim`, `ylim` numeric vectors of length 2, giving the x and y coordinates ranges. if they equal to `NULL` then they will be adjusted from the data.
- `pch` plotting character, i.e., symbol to use in the 3-D scatter plot.
- `textcex` a numerical value giving the amount by which plotting text should be magnified relative to the default.
- `plotcex` a numerical value giving the amount by which plotting symbols should be magnified relative to the default.
- `angle` angle between x and y axis.
- `col` color of the text.
- `col.regions` color vector to be used reflecting magnitude of the dataset at different locations. The general idea is that this should be a color vector of gradually varying color.
- `...` further arguments passed to plot and panel settings.
Author(s)
Zifei Han <hanzifei@gmail.com>

See Also
plot.mlegc, plot.predgc

plotgc
Plot Geostatistical Count Data

Description
This function generates two plots describing a geostatistical count data. The first plot is a bubble plot with size proportional to the response. The second plot is a lattice plot with text describing the number of counts.

Usage
plotgc(data = NULL, locs = NULL, bdry = NULL, col = 2, pch = 1,
       textcex = 1, col.regions = gray(90:0/100), size = c(0.3, 2.7), ...)

Arguments
- data: the geostatistical count response.
- locs: a n by 2 matrix or data frame that indicates the coordinates of locations.
- bdry: a list containing the coordinates of boundaries.
- col: the color used for response variable in both plots.
- pch: the shape used for response variable in the first plot.
- textcex: a numerical value giving the amount by which plotting text should be magnified relative to the default.
- col.regions: color vector to be used reflecting magnitude of the dataset at different locations. The general idea is that this should be a color vector of gradually varying color.
- size: the minimum and maximum of the sizes in the first plot.
- ...: other parameters that control the plotting.

Author(s)
Zifei Han <hanzifei@gmail.com>

See Also
plot.simgc, plot.mlegc, plot.predgc
### Poisson Marginal of Class `marginal.gc`

**Description**

The Poisson marginal parameterized in terms of its mean.

By default, this function is used for likelihood inference and spatial prediction in function `mlegc` and `predgc` of the package `gcKrig`. When all marginal parameters are given, the function is used for simulation and computing correlation in a trans-Gaussian random field in function `simgc` and `corrtG`.

**Usage**

```r
gcPoissonH(link = "log", lambda = NULL)
```

**Arguments**

- `link`: the model link function.
- `lambda`: a non-negative scalar of the mean parameter.

**Value**

An object of class `marginal.gc` representing the marginal component.

**Author(s)**

Zifei Han <hanzifei1@gmail.com>

**See Also**

- `marginal.gc`, `beta.gc`, `binomial.gc`, `gm.gc`, `gaussian.gc`, `negbin.gc`, `weibull.gc`, `zip.gc`

---

### Powered Exponential Correlation Function of Class `corr.gc`

**Description**

The powered exponential correlation function in spatial statistics.

By default, range parameter is not available, so this function is used for likelihood inference and spatial prediction in function `mlegc` and `predgc`. Users need to specify the shape parameter `kappa` and if the correlation model includes a nugget effect `nugget = TRUE` or not `nugget = FALSE`.

When both range and nugget parameters are given, the function is used for simulation with function `simgc` in package `gcKrig`. 

---
Usage

`powerexp.gc(range = NULL, kappa = 1, nugget = TRUE)`

Arguments

- `range`: a non-negative scalar of the range parameter in powered exponential correlation function.
- `kappa`: a scalar between 0 and 2; the value of the shape parameter in the powered exponential correlation function.
- `nugget`: the nugget effect of the correlation function. If specified, it must be a scalar between 0 and 1.

Details

The powered exponential correlation function with a nugget $\tau^2$ is of the form:

$$\rho(h) = (1 - \tau^2) \exp((-h/\phi)^\kappa)$$

when $h > 0$ and $\rho(h) = 1$ when $h = 0$. Here $h$ is distance, $\phi$ is range parameter, $\kappa$ is the shape parameter and $\tau^2$ is the nugget effect.

When using the powered exponential correlation function, note that $0 < \kappa \leq 2$.

Value

An object of class `corr.gc` representing the correlation component.

Author(s)

Zifei Han <hanzifei1@gmail.com>

See Also

`matern.gc`, `spherical.gc`

Description

Computes the plug-in prediction at unobserved sites. Two methods are implemented. If `method = 'GHK'` then the maximum simulated likelihood estimates are computed and the sequential importance sampling method is used in the integral evaluation. If `method = 'GQT'` then the maximum surrogate likelihood estimates are computed and the generalized quantile transform method is used in integral approximation.
Usage

predgc(obs.y, obs.x = NULL, obs.locs, pred.x = NULL, pred.locs, 
longlat = FALSE, distscale = 1, marginal, corr, obs.effort = 1, 
pred.effort = 1, method = "GHK", estpar = NULL, corrpar0 = NULL, 
pred.interval = NULL, parallel = FALSE, 
ghkoptions = list(nrep = c(100, 1000), reorder = FALSE, seed = 12345), 
paralleloptions = list(n.cores = 2, cluster.type = "SOCK"))

Arguments

obs.y  a non-negative integer vector of observed response with its length equals to the number of observed locations.

obs.x  a numeric matrix or data frame of covariates at observed locations, with its number of rows equals to the number of observed locations. If no covariates then obs.x = NULL.

obs.locs  a numeric matrix or data frame of observed locations. obs.effort The first column is x or longitude, the second column is y or latitude. The number of observed locations is equal to the number of rows.

pred.x  a numeric matrix or data frame of covariates at prediction locations, with its number of rows equals to the number of prediction locations. If no covariates then pred.x = NULL.
	pred.locs  a numeric matrix or data frame of prediction locations. First column is x or longitude, second column is y or latitude. The number of prediction locations equals to the number of rows.

longlat  if FALSE, use Euclidean distance, if TRUE use great circle distance. The default is FALSE.

distscale  a numeric scaling factor for computing distance. If original distance is in kilometers, then distscale = 1000 will convert it to meters.

marginal  an object of class marginal.gc specifying the marginal distribution.

corr  an object of class corr.gc specifying the correlation function.

obs.effort  sampling effort at observed locations. For binomial marginal it is the size parameter (number of trials). See details.

pred.effort  sampling effort at prediction locations. For binomial marginal it is the size parameter (number of trials). See details.

method  two methods are implemented. If method = ‘GHK’ then the maximum simulated likelihood estimates are computed, if method = ‘GQT’ then the maximum surrogate likelihood estimates are computed.

estpar  if estpar = NULL then the likelihood estimates will be computed first, then plug-in into the predictive density. When all estimates are available, it is suggested to specify estpar with the parameter estimates so re-fitting is not needed. If so, the sequence of the input values for estpar is: regression parameters, overdispersion (if any), and correlation parameters (range and nugget, if applicable).
**corrpar0** the starting value of correlation parameters in optimization procedure. If \( \text{corrpar0} = \text{NULL} \) then initial range is set to be half of the median distance in distance matrix, and initial nugget (if nugget = TRUE) is 0.2.

**pred.interval** a number between 0 and 1 representing confidence level of the prediction interval. The program will output two types of the prediction intervals, see detail. If \( \text{pred.interval} = \text{NULL} \) then no prediction interval will be computed.

**parallel** if TRUE then parallel computing is used to predict multiple prediction locations simultaneously. If FALSE then a serial version will be called.

**ghkoptions** a list of three elements that only need to be specified if \( \text{method} = \text{GHK} \).

\( \text{nrep} \) is the Monte Carlo size of the importance sampling algorithm for likelihood approximation. It can be a vector with increasing positive integers so that the model is fitted with a sequence of different Monte Carlo sizes, and the starting values for optimization are taken from the previous fitting. The default value is 100 for the first optimization and 1000 for the second and definitive optimization.

**reorder** indicates whether the integral will be reordered every iteration in computation according to the algorithm in Gibson, et al (1994), default is FALSE.

**seed** is seed of the pseudorandom generator used in Monte Carlo simulation.

**parallel.options** a list of two elements that only need to be specified if \( \text{parallel} = \text{TRUE} \).

\( \text{n.cores} \) is the number of cores to be used in parallel prediction.

\( \text{cluster.type} \) is type of cluster to be used for parallel computing; can be "SOCK", "MPI", "PVM", or "NWS".

**Details**

This program implemented two methods in predicting the response at unobserved sites. See \text{mlegc}.

The argument \text{obs.effort} and \text{pred.effort} are the sampling effort (known). It can be used to consider heterogeneity of the measurement time or area at different locations. The default is 1 for all locations. See Han and De Oliveira (2016) for more details.

The program computes two types of prediction intervals at a given confidence level. The shortest prediction interval is obtained from evaluating the highest to lowest prediction densities; the equal tail prediction interval has equal tail probabilities.

**Value**

A list of class "predgc" with the following elements:

- **obs.locs** observed locations.
- **obs.y** observed values at observed locations.
- **pred.locs** prediction locations.
- **predValue** the expectation of the conditional predictive distribution.
- **predCount** predicted counts; the closest integer that \text{predValue} rounded to.
- **predVar** estimated variance of the prediction at prediction locations.
predgc

ConfidenceLevel

confidence level (between 0 to 1) if prediction interval is computed.

predInterval.EqualTail

equal-tail prediction interval.

predInterval.Shortest

shortest length prediction interval.

Author(s)

Zifei Han <hanzifei1@gmail.com>

References


See Also

gcmr; mlegc

Examples

```r
## Not run:
## For fast check predict at four locations only
#data(Weed95)
weedobs <- Weed95[Weed95$dummy==1, ]
weedpred <- Weed95[Weed95$dummy==0, ]
predweed1 <- predgc(obs.y = weedobs$weedcount, obs.x = weedobs[,4:5], obs.locs = weedobs[,1:2],
  pred.x = weedpred[,1:4,4:5], pred.locs = weedpred[,1:4,1:2],
  marginal = negbin.gc(link = 'log'), pred.interval = 0.9,
  corr = matern.gc(kappa = 0.5, nugget = TRUE), method = 'GHK')
#summary(predweed1)
#plot(predweed1)

## Time consuming examples
## Weed prediction at 200 locations using parallel programming
predweed2 <- predgc(obs.y = weedobs$weedcount, obs.x = weedobs[,4:5], obs.locs = weedobs[,1:2],
  pred.x = weedpred[,4:5], pred.locs = weedpred[,1:2],
```
profile.mlegc

Profile Likelihood Based Confidence Interval of Parameters for Gaussian Copula Models in Geostatistical Count Data

Description

This function computes the (approximate) profile likelihood based confidence interval. The algorithm starts by choosing two starting points at different sides of the MLE and using an iterative process to find the approximate lower and upper bound.
Usage

```r
## S3 method for class 'mlegc'
profile(fitted, par.index, alpha = 0.05, start.point = NULL,
         method = 'GQT', nrep = 1000, seed = 12345, ...)
```

Arguments

- `fitted`: an object of class `mlegc`, typically inherited from function `mlegc`.
- `par.index`: the index of the parameter which should be profiled.
- `alpha`: the significance level, default is `0.05` which corresponds to 95 percent confidence interval.
- `start.point`: numeric vector of length 2 indicating the starting points for finding the left and right bound. If `start.point = NULL` then the default starting points will be used.
- `method`: Two methods are implemented. If `method = 'GHK'` then the simulated likelihood will be used, if `method = 'GQT'` then the surrogate likelihood will be used.
- `nrep`: the Monte Carlo size of the importance sampling algorithm for likelihood approximation; only need to be specified if `method = 'GHK'`.
- `seed`: seed of the pseudorandom generator used in Monte Carlo simulation; only need to be specified if `method = 'GHK'`.
- `...`: other arguments passed.

Value

Lower and upper bounds of the approximate confidence interval.

Author(s)

Zifei Han <hanzifei1@gmail.com>

References


See Also

`mlegc`
Examples

```r
## Not run:
data(LansingTrees)
Treefit4 <- mlegc(y = LansingTrees[,3], x = LansingTrees[,4],
  locs = LansingTrees[,1:2], marginal = zip.gc(link = 'log'),
  corr = matern.gc(kappa = 0.5, nugget = TRUE), method = 'GHK')
summary(Treefit4)
profile(Treefit4, 1, 0.05, method = 'GHK', nrep = 1000, seed = 12345)
profile(Treefit4, 2, 0.05, method = 'GHK', nrep = 1000, seed = 12345)
profile(Treefit4, 3, 0.05, method = 'GHK', nrep = 1000, seed = 12345)
profile(Treefit4, 4, 0.05, method = 'GHK', nrep = 1000, seed = 12345)
profile(Treefit4, 5, 0.05, method = 'GHK', nrep = 1000, seed = 12345)
## End(Not run)
```

Description

Simulate geostatistical data from Gaussian copula model at given locations. This function can simulate multiple datasets simultaneously.

Usage

```r
simgc(locs, sim.n = 1, marginal, corr, longlat = FALSE)
```

Arguments

- **locs**: a numeric matrix or data frame of \( n \)-\( D \) points with row denoting points. First column is \( x \) or longitude, second column is \( y \) or latitude. The number of locations is equal to the number of rows.
- **sim.n**: the number of simulation samples required.
- **marginal**: an object of class `marginal.gc` specifying the marginal distribution.
- **corr**: an object of class `corr.gc` specifying the correlation function.
- **longlat**: if FALSE, use Euclidean distance, if TRUE use great circle distance. Default is FALSE.

Value

A list of two elements:
- **data**: a numeric matrix with each row denoting a simulated data.
- **locs**: the location of the simulated data, same as the input locs.
Author(s)

Zifei Han <hanzifei1@gmail.com>

Examples

```r
grid <- seq(0.05, 0.95, by = 0.1)
xloc <- expand.grid(x = grid, y = grid)[,1]
yloc <- expand.grid(x = grid, y = grid)[,2]
set.seed(12345)
sim1 <- simgc(locs = cbind(xloc, yloc), sim.n = 10, marginal = negbin.gc(mu = 5, od = 1),
               corr = matern.gc(range = 0.3, kappa = 0.5, nugget = 0.1))
#plot(sim1, index = 1)
```

The Spherical Correlation Function of Class `corr.gc`

Description

The spherical correlation function in spatial statistics.

By default, range parameter is not available, so this function is used for likelihood inference and spatial prediction in function `mlegc` and `predgc`. Users need to specify if the correlation model includes a nugget effect `nugget = TRUE` or not `nugget = FALSE`.

When both range and nugget parameters are given, the function is used for simulation with function `simgc` in package `gcKrig`.

Usage

```r
spherical.gc(range = NULL, nugget = TRUE)
```

Arguments

- `range` a non-negative scalar of the range parameter in the spherical correlation function.
- `nugget` the nugget effect of the correlation function. If specified, it must be a scalar between 0 and 1.

Details

The spherical correlation function with a nugget $\tau^2$ is of the form:

$$
\rho(h) = (1 - \tau^2)(1 - 1.5(h/\phi) + 0.5(-h/\phi)^3)
$$

when $h > 0$ and $\rho(h) = 1$ when $h = 0$, $h$ is distance.

Value

An object of class `corr.gc` representing the correlation component.
Methods for Extracting Information from Fitted Object of Class \texttt{mlegc}

\section*{Description}

Return a summary table of results from model fitting.

\section*{Usage}

\begin{verbatim}
## S3 method for class 'mlegc'
summary(object, ...)
\end{verbatim}

\section*{Arguments}

\begin{itemize}
\item \texttt{object} an object of class \texttt{mlegc} inherited from function \texttt{mlegc}.
\item \texttt{...} additional arguments, but currently not used.
\end{itemize}

\section*{Value}

A table summary of the estimates, standard error, z-value and several information criteria.

\section*{Author(s)}

Zifei Han \texttt{<hanzifei1@gmail.com>}

\section*{See Also}

\texttt{mlegc}
Methods for Extracting Information from Fitted Object of Class `predgc`

Description

Output a summary data frame.

Usage

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

Arguments

- `object`: an object of class `predgc` inherited from function `predgc`.
- `...`: further arguments.

Value

A table including the following information:

- `pred.locs`: prediction locations.
- `predMean`: the expectation of the conditional predictive distribution.
- `predCount`: predicted counts; the closest integer that `predMean` rounded to.
- `predVar`: estimated variance of the prediction at prediction locations.
- `predInterval.EqualTail`: equal-tail prediction interval; computed only if `ConfidenceLevel = TRUE`.
- `predInterval.Shortest`: shortest length prediction interval; computed only if `ConfidenceLevel = TRUE`.

Author(s)

Zifei Han <hanzifei@gmail.com>

See Also

- `mlegc`,`predgc`
Weed95

Counts of Weed Plants on a Field

Description

The weed species Viola Arvensis was counted within circular frames each of area 0.25 square meter except for 10 missing sites in the first row, from a 20 by 14 rectangular grid, so the total number of locations is 270. Also, the percentages of organic matter in a soil sample are collected. The data was studied by Christensen and Waagepetersen (2002) to investigate whether weed occurrence could be predicted from observations of soil texture and soil chemical properties.

Usage

data(Weed95)
Format

A data frame with 270 observations and 6 variables.

- xloc Cartesian x-coordinate of the locations (in meter).
- yloc Cartesian y-coordinate of the locations (in meter).
- weedcount Number of weed collected at the given site.
- scaled Y coord The scaled Y coordinate with range -1 to 1 as a covariate in regression.
- organic Another chemical component indicating the organic matter of the soil.
- dummy A dummy variable taking values 0 or 1. If 0 it is treated as observed location and 1 treated as predicted location in Christensen and Waagepetersen (2002).

References


Examples

data(Weed95)
str(Weed95)

weibull.gc

The Weibull Marginal of Class marginal.gc

Description

The Weibull marginal used for simulation and computing correlation in the trans-Gaussian random field in function simgc and corrTG of the package gckrig. It cannot be used in function mlegec nor predgc to make model inferences.

Usage

weibull.gc(shape = 1, scale = 1)

Arguments

- shape a positive scalar of shape parameter in the Weibull distribution.
- scale a positive scalar of scale parameter in the Weibull distribution.

Details

The Weibull distribution with shape parameter $a$ and scale parameter $b$ has density given by

$$(a/b)(x/b)^{a-1}exp(-(x/b)^a)$$
zip.gc

Value
An object of class `marginal.gc` representing the marginal component.

Author(s)
Zifei Han <hanzifei1@gmail.com>

See Also
`marginal.gc`, `beta.gc`, `binomial.gc`, `gm.gc`, `gaussian.gc`, `negbin.gc`, `poisson.gc`, `zip.gc`

zip.gc

The Zero-inflated Poisson Marginal of Class `marginal.gc`

Description
The zero-inflated Poisson marginal parameterized in terms of its mean and overdispersion.

By default, this function is used for likelihood inference and spatial prediction in function `mlegc` and `predgc` of the package `gckrig`. When all marginal parameters are given, the function is used for simulation and computing correlation in a trans-Gaussian random field in function `simgc` and `corrTG`.

Usage
zip.gc(link = "log", mu = NULL, od = NULL)

Arguments
- `link`: the model link function.
- `mu`: a non-negative scalar of the mean parameter.
- `od`: a non-negative scalar of the overdispersion parameter.

Details
The zero-inflated Poisson distribution with parameters \( \mu = a \) and \( od = b \) has density

\[
b/(1 + b) + \exp(-(a + ab))/(1 + b)
\]

when \( y = 0 \), and

\[
\exp(-(a + ab)) * (a + ab)^y / ((1 + b)y!)
\]

when \( y = 1, 2, \ldots \)

Under this parameterization, \( \text{var}(Y) = \mu + od * \mu^2 \), where \( \mu \) is the mean parameter and \( od \) is the overdispersion parameter. For more details see Han and De Oliveira (2016).
Value

An object of class `marginal.gc` representing the marginal component.

Author(s)

Zifei Han <hanzifei1@gmail.com>

References


See Also

`marginal.gc, beta.gc, binomial.gc, gm.gc, gaussian.gc, negbin.gc, poisson.gc, weibull.gc`
Index

* Topic **Correlation Structure**
corrTG, 6
FHubDiscrete, 7
* Topic **Correlation**
corr.gc, 5
matern.gc, 12
powerexp.gc, 25
spherical.gc, 33

* Topic **Data Visualization**
plot.mlegc, 21
plot.predgc, 22
plot.simgc, 23
plotgc, 24

* Topic **Datasets**
AtlanticFish, 2
LansingTrees, 10
OilWell, 20
Weed95, 36

* Topic **Estimation**
marginal.gc, 11
mlegc, 13
mvnintGKH, 17
plot.mlegc, 21
profile.mlegc, 30
summary.mlegc, 34
vcov.mlegc, 36

* Topic **Marginal**
beta.gc, 3
binomial.gc, 4
gaussian.gc, 8
gm.gc, 9
marginal.gc, 11
negbin.gc, 19
poisson.gc, 25
weibull.gc, 37
zip.gc, 38

* Topic **Prediction**
marginal.gc, 11
plot.predgc, 22
predgc, 26
summary.predgc, 35

* Topic **Simulation**
beta.gc, 3
binomial.gc, 4
corr.gc, 5
gaussian.gc, 8
gm.gc, 9
marginal.gc, 11
matern.gc, 12
negbin.gc, 19
plot.simgc, 23
poisson.gc, 25
powerexp.gc, 25
simgc, 32
spherical.gc, 33
weibull.gc, 37
zip.gc, 38

AtlanticFish, 2
beta.gc, 3, 4, 9, 11, 12, 20, 25, 38, 39
binomial.gc, 4, 4, 9, 11, 12, 20, 25, 38, 39
corr.gc, 5, 12–14, 25–27, 32, 33
corrTG, 6

FHubDiscrete, 7

gaussian.gc, 4, 8, 9, 11, 12, 20, 25, 38, 39
gcmr, 15, 29
gm.gc, 4, 9, 11, 12, 20, 25, 38, 39

LansingTrees, 10

marginal.gc, 3, 4, 6, 8, 9, 11, 14, 19, 20, 25, 27, 32, 37–39
matern.gc, 5, 12, 26, 34
mlegc, 13, 21, 23, 28, 29, 31, 34–36
mvnintGKH, 17
negbin.gc, 4, 9, 11, 12, 19, 25, 38, 39

OilWell, 20

plot.mlegg, 21, 23, 24
plot.predgc, 21, 22, 24
plot.simgc, 21, 23, 23, 24
plotgc, 24
pmvnorm, 18
poisson.gc, 4, 9, 11, 12, 20, 25, 38, 39
powerexp.gc, 5, 13, 25
predgc, 22, 23, 26, 35
print.mlegg (summary.mlegg), 34
print.summary.mlegg (summary.mlegg), 34
profile.mlegg, 30

simgc, 23, 32
spherical.gc, 5, 13, 26, 33, 34
summary.mlegg, 34
summary.predgc, 35

vcov.mlegg, 36

Weed95, 36
weibull.gc, 4, 9, 11, 12, 20, 25, 37, 39

zip.gc, 4, 9, 11, 12, 20, 25, 38, 38