

# Package ‘usdm’

June 25, 2017

**Type** Package

**Title** Uncertainty Analysis for Species Distribution Models

**Version** 1.1-18

**Date** 2017-06-25

**Author** Babak Naimi

**Depends** methods, sp, raster

**Suggests** rgdal (>= 0.6-28)

**Maintainer** Babak Naimi <naimi.b@gmail.com>

**Description** This is a framework that aims to provide methods and tools for assessing the impact of different sources of uncertainties (e.g. positional uncertainty) on performance of species distribution models (SDMs.)

**License** GPL (>= 3)

**URL** <http://r-gis.net>

**Repository** CRAN

**Repository/R-Forge/Project** usdm

**Repository/R-Forge/Revision** 27

**Repository/R-Forge/DateTimeStamp** 2017-06-25 15:16:10

**Date/Publication** 2017-06-25 16:03:49 UTC

**NeedsCompilation** no

## R topics documented:

usdm-package . . . . .	2
exclude . . . . .	2
lisa . . . . .	4
plot.RasterVariogram . . . . .	6
plot.speciesLISA . . . . .	7
RasterVariogram-class . . . . .	9
speciesLisa . . . . .	10
speciesLISA-class . . . . .	12

Variogram . . . . .	12
vif . . . . .	14
VIF-class . . . . .	16

<b>Index</b>	<b>17</b>
--------------	-----------

---

usdm-package	<i>Uncertainty analysis for SDMs</i>
--------------	--------------------------------------

---

### Description

This package provides a number of functions for exploring the impact of different sources of uncertainties (e.g. positional uncertainty) on performance of species distribution models (SDMs).

In addition, there is a function to quantify different local indicators of spatial association (LISA) for raster data.

### Author(s)

Babak Naimi <naimi.b@gmail.com>

<http://r-gis.net>

---

exclude	<i>Excluding variables specified in a VIF object</i>
---------	--

---

### Description

Physically exclude the collinear variables which are identified using `vifcor` or `vifstep` from a set of variables.

### Usage

```
exclude(x, vif, ...)
```

### Arguments

<code>x</code>	explanatory variables (predictors), defined as a raster object ( <code>RasterStack</code> or <code>RasterBrick</code> ), or as a matrix, or as a <code>data.frame</code> .
<code>vif</code>	an object of class <code>VIF</code> , resulted from <code>vifcor</code> or <code>vifstep</code> functions.
<code>...</code>	additional argument as in <code>vifstep</code>

### Details

Before using this function, you should execute one of `vifstep` or `vifcor` which detect collinearity based on calculating variance inflation factor (VIF) statistics. If `vif` is missing, then `vifstep` is called.

**Value**

an object of class same as x (i.e. RasterStack or RasterBrick or data.frame or matrix)

**Author(s)**

Babak Naimi <naimi.b@gmail.com>

<http://r-gis.net>

**References**

IF you used this method, please cite the following article for which this package is developed:

Naimi, B., Hamm, N.A.S., Groen, T.A., Skidmore, A.K., and Toxopeus, A.G. 2014. Where is positional uncertainty a problem for species distribution modelling?, *Ecography* 37 (2): 191-203.

**See Also**

[vif](#)

**Examples**

```
## Not run:
file <- system.file("external/spain.grd", package="usdm")

r <- brick(file) # reading a RasterBrick object including 10 raster layers in Spain

r

vif(r) # calculates vif for the variables in r

v1 <- vifcor(r, th=0.9) # identify collinear variables that should be excluded

v1

re1 <- exclude(r,v1) # exclude the collinear variables that were identified in
# the previous step

re1

v2 <- vifstep(r, th=10) # identify collinear variables that should be excluded

v2

re2 <- exclude(r, v2) # exclude the collinear variables that were identified in
# the previous step

re2

re3 <- exclude(r) # first, vifstep is called

re3
```

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

---

`lisa`

*Local indicators of spatial association*

---

### Description

Calculate different statistics of local indicator of spatial association (LISA) for each cell in a raster data.

### Usage

```
lisa(x, y, d1=0, d2, cell, statistic="I")
```

### Arguments

<code>x</code>	a raster object (RasterLayer or RasterStack or RasterBrick)
<code>y</code>	a SpatialPoints object (optional)
<code>d1</code>	numeric. A number (distance), specifies local neighborhood size. Default is 0, means that the local neighborhood starts from the cell (distance = 0) and ends to a distance = d2
<code>d2</code>	numeric. A number (distance), specifies local neighborhood size. It specifies the distance to which should be considered as a local neighborhood around a cell
<code>cell</code>	numeric (optional). A cell number or a vector of cell numbers in the Raster object, at which LISA should be calculated
<code>statistic</code>	a character string specifying the LISA statistic that should be calculated. This can be one of "I", "c", "G", "G*", and "K1"

### Details

This function can calculate different LISA statistics at each grid cell in Raster object. The statistics, implemented in this function, include local Moran's I ("I"), local Geary's c ("c"), local G and G\* ("G" and "G\*"), and local K1 statistics. This function returns standardized value (Z) for Moran, G and G\*, and K1 statistics. If a SpatialPoints or a vector of numbers is defined for y or cell, the LISA is calculated only for the specified locations by points or cells.

### Value

RasterLayer	if x is a RasterLayer and both y and cell are missed
RasterBrick	if x is a RasterStack or a RasterBrick and both y and cell are missed
numeric vector	if y or cell is specified

**Author(s)**

Babak Naimi <naimi.b@gmail.com>

<http://r-gis.net>

**References**

Anselin, L. 1995. Local indicators of spatial association, *Geographical Analysis*, 27, 93–115;

Getis, A. and Ord, J. K. 1996 Local spatial statistics: an overview. In P. Longley and M. Batty (eds) *Spatial analysis: modelling in a GIS environment* (Cambridge: Geoinformation International), 261–277.

Naimi, B., Hamm, N.A.S., Groen, T.A., Skidmore, A.K., and Toxopeus, A.G. 2014. Where is positional uncertainty a problem for species distribution modelling?, *Ecography* 37 (2): 191-203.

**Examples**

## Not run:

```
file <- system.file("external/spain.grd", package="usdm")
```

```
r <- brick(file) # reading a RasterBrick object including 10 rasters in Spain
```

```
r
```

```
plot(r) # visualize the raster layers
```

```
plot(r[[1]]) # visualize the first raster layer
```

```
r.I <- lisa(x=r[[1]],d1=0,d2=25000,statistic="I") # local Moran's I
```

```
plot(r.I)
```

```
# entering r instead of r[[1]], gives the indicator for each layer:
```

```
r.I <- lisa(x=r,d1=0,d2=25000,statistic="I")
```

```
plot(r.I)
```

```
r.c <- lisa(x=r[[1]],d1=0,d2=25000,statistic="c") # local Geary's c
```

```
plot(r.c)
```

```
r.g <- lisa(x=r[[1]],d1=0,d2=25000,statistic="G") # G statistic
```

```
plot(r.g)
```

```
r.g2 <- lisa(x=r[[1]],d1=0,d2=25000,statistic="G*") # G* statistic
```

```
plot(r.g2)
```

```
r.K1 <- lisa(x=r[[1]],d1=0,d2=30000,statistic="K1") # gives K1 statistic for each layer
```

```

plot(r.K1)

lisa(x=r,d1=0,d2=30000,cell=2000,statistic="I") # gives local Moran's I at cell number 2000
#for each raster layer in r

lisa(x=r,d1=0,d2=30000,cell=c(2000,2002,2003),statistic="c") # calculates local Moran's I
# at cell numbers of 2000,2002, and 2003 for each raster layer in r

sp <- sampleRandom(r[[1]],20,sp=TRUE) # draw 20 random points from r,
# and returns a SpatialPointsDataFrame

plot(r[[1]])

points(sp)

lisa(x=r,y=sp,d1=0,d2=30000,statistic="I") # calculates the local Moran's I at
# point locations in sp for each raster layer in r

## End(Not run)

```

---

`plot.RasterVariogram` *Plot variogram or variogram cloud or boxplot based on variogram cloud*

---

### Description

Plot the variogram computed for raster data by Variogram function

### Usage

```

## S4 method for signature 'RasterVariogram'
plot(x, ...)

```

### Arguments

<code>x</code>	an object of class RasterVariogram, which is the output of <a href="#">Variogram</a> function.
<code>...</code>	additional argument (see details)

### Details

This function plot the empirical variogram, or variogram cloud if `cloud` set to TRUE or a boxplot of variogram cloud data if `box` set to TRUE,

Below are additional arguments:

cloud logical. If TRUE, the function plots variogram cloud. box logical. If TRUE, the function plots boxplot of variogram cloud. ... xlab, ylab and main and other arguments are same as the base plot function.

**Value**

plots the variogram.

**Author(s)**

Babak Naimi <naimi.b@gmail.com>

<http://r-gis.net>

**See Also**

[Variogram](#)

**Examples**

```
file <- system.file("external/spain.grd", package="usdm")
r <- brick(file) # reading a RasterBrick including 5 rasters (predictor variables)
r
plot(r[[1]]) # visualize the raster layers
v1 <- Variogram(r[[1]]) # compute variogram for the first raster

plot(v1)
plot(v1,cloud=TRUE)
plot(v1,box=TRUE)
```

---

plot.speciesLISA

*Plot positional uncertainty based on LISA*

---

**Description**

Plot the values of LISAs at species occurrence locations, which can be used to identify the locations that need positional uncertainty treatment.

## Usage

```
## S4 method for signature 'speciesLISA,missing'  
plot(x, y, ...)  
## S4 method for signature 'speciesLISA,SpatialPolygons'  
plot(x, y, ...)  
## S4 method for signature 'speciesLISA,SpatialPolygonsDataFrame'  
plot(x, y, ...)
```

## Arguments

**x** an object of class speciesLISA, which is the output of [speciesLisa](#) function.  
**y** optional. Boundary map of the study area, an object of class SpatialPolygons.  
**...** additional argument (see details)

## Details

This function generates a map (i.e. a bubble plot) in which the species points present the magnitude of LISA in predictors at the location as open or filled circles with different sizes.

Below are additional arguments:

**cex** the maximum symbol size (circle) in the plot. **levels** specifies the number of LISA levels at which the points are presented. **xyLegend** a vector with two numbers, specifying the coordinates of the legend. If missing, the function tries to find the appropriate location for it. **...** **xlab**, **ylab** and **main** same as the base plot function.

## Value

plots the bubble plot.

## Author(s)

Babak Naimi <naimi.b@gmail.com>  
<http://r-gis.net>

## See Also

[speciesLisa](#), [lisa](#)

## Examples

```
file <- system.file("external/predictors.grd", package="usdm")  
  
r <- brick(file) # reading a RasterBrick including 5 rasters (predictor variables)  
  
r  
  
plot(r) # visualize the raster layers
```



```
#files.path <- system.file("external", package="usdm") # path to location of example files
#require(rgdal)
#sp <- readOGR(dsn=files.path,layer="species_nl") # reading species data (shapefile)

#splisa <- speciesLisa(x=r,y=sp,uncertainty=15000,weights=c(0.22,0.2,0.38,0.2))

#splisa

#plot(splisa)

#bnd <- readOGR(dsn=files.path,layer="boundary") # reading the boundary map

#plot(splisa,bnd)

#plot(splisa,bnd,levels=c(2,4,6,8))

#plot(splisa,bnd,levels=c(-5,-3,0,3,5))
```

---

RasterVariogram-class *RasterVariogram class*

---

## Description

An object of the RasterVariogram class contains information about the empirical variogram of a raster data. The object can be created with the function: [Variogram](#).

## Slots

Slots for speciesLISA object:

lag: a number specifying lag distance

nlags: a number specifying number of lags based on cutoff parameter

variogramCloud: matrix, including semivariance for all pairs

variogram: data.frame, including binned semivariance within each lag

## Author(s)

Babak Naimi <naimi@r-gis.net>

<http://r-gis.net>

## Examples

```
showClass("speciesLISA")
```

---

 speciesLisa

*LISA in predictors at species occurrence locations*


---

## Description

Given a level of positional uncertainty (defined as a distance), this function calculates different statistics of local indicator of spatial association (LISA) in predictors (explanatory variables, defined as a raster object) at each species occurrence location (defined as a SpatialPoints object). According to Naimi et al. 2012, this can be used to understand whether positional uncertainty at which species locations are likely to affect predictive performance of species distribution models.

## Usage

```
speciesLisa(x, y, uncertainty, statistic="K1", weights)
```

## Arguments

x	explanatory variables (predictors), defined as a raster object (RasterLayer or RasterStack or RasterBrick)
y	species occurrence points, defined as a SpatialPoints or SpatialPointsDataFrame object
uncertainty	level of positional uncertainty, defined as a number (distance)
statistic	a character string specifying the LISA statistic that should be calculated. This can be one of "I", "c", "G", "G*", and "K1". Default is "K1"
weights	a numeric vector specifying the relative importance of explanatory variables in species distribution models (the first value in the weights, is the importance of the first variable in x, and ...). These values will be used as weights to aggregate the LISAs in predictors at each location and calculate a single measure. The length of weights should be equal to the number of raster layers in x

## Details

This function calculates a LISA statistic for each explanatory variable at each species point. Although several statistics including local Moran's I ("I"), local Geary's c ("c"), local G and G\* ("G" and "G\*"), and local K1 statistics, can be calculated, according to Naimi et al. (2012), "K1" statistic (default) is recommended. This function returns a speciesLISA object, which includes species occurrence data, LISA statistic for each predictor at species locations, and an aggregated LISA statistic (a single LISA) at each species location, given the variable importances. If weights in not specified, the equal weights (i.e. equal importance for explanatory variables) will be considered.

## Value

speciesLISA

**Author(s)**

Babak Naimi <naimi.b@gmail.com>  
<http://r-gis.net>

**References**

IF you used this method, please cite the following article for which this package is developed:

Naimi, B., Hamm, N.A.S., Groen, T.A., Skidmore, A.K., and Toxopeus, A.G. 2014. Where is positional uncertainty a problem for species distribution modelling?, *Ecography* 37 (2): 191-203.

**See Also**

[lisa](#)

**Examples**

```
## Not run:
file <- system.file("external/predictors.grd", package="usdm")

r <- brick(file) # reading a RasterBrick object including 4 rasters in the Netherlands

r

plot(r) # visualize the raster layers

files.path <- system.file("external", package="usdm") # path to location of example files
require(rgdal)
sp <- readOGR(dsn=files.path,layer="species_nl") # reading species data (shapefile)

# alternatively, to read species data you can use:
require(maptools)
sp.file <- system.file("external/species_nl.shp", package="usdm")
sp <- readShapePoints(sp.file)

splisa <- speciesLisa(x=r,y=sp,uncertainty=15000,weights=c(0.22,0.2,0.38,0.2))

splisa

plot(splisa)

bnd <- readOGR(dsn=files.path,layer="boundary") # reading the boundary map

plot(splisa,bnd)

## End(Not run)
```

---

speciesLISA-class      *speciesLISA class*

---

### Description

An object of the speciesLISA class contains information about a local indicator of spatial association (LISA) statistic in predictor variables at the location of species occurrences. The object can be created with the function: [speciesLisa](#).

### Slots

Slots for speciesLISA object:

species: object of class SpatialPoints

data: data.frame, attribute table of species points

LISAs: matrix, LISA statistics for different predictors

weights: numeric, the variable importance

statistic: character, the name of LISA statistic

LISA: numeric, aggregated LISAs at each species location

### Author(s)

Babak Naimi <naimi.b@gmail.com>

<http://r-gis.net>

### Examples

```
showClass("speciesLISA")
```

---

Variogram

*Empirical variogram for raster data*

---

### Description

Compute sample (empirical) variogram from raster data. The function returns a binned variogram and a variogram cloud.

### Usage

```
Variogram(x, lag, cutoff, cells, size=100)
```

**Arguments**

x	a raster object (RasterLayer)
lag	the lag size (width of subsequent distance intervals) into which cell pairs are grouped for semivariance estimates. If missing, the cell size (raster resolution) is assigned.
cutoff	spatial separation distance up to which cell pairs are included in semivariance estimates; as a default, the length of the diagonal of the box spanning the data is divided by three.
cells	numeric (optional). A vector of cell numbers in the Raster object. This forces the function to only consider these cells (and their neighbours) to compute the variogram.
size	positive integer specifying the number of cells to be drawn from raster object. If the number of cells in the raster object is large, a sample with the specified size is drawn to make the computation more efficient.

**Details**

Variograms are widely used for exploring spatial structure in a single variable. Formally, it is defined as half the expected squared difference (half the variance of the difference) in the variable value at a specific geographical separation. A variogram summarizes the spatial relations in the data, and can be used to understand within what range (distance) the data is spatially autocorrelated. Naimi et al. (2011) linked this range to the impact of positional uncertainty on the performance of species distribution models (SDMs). Based on that study, examining variogram to find the effective autocorrelation range in predictors gives insight into whether predictions by SDMs are likely to be affected by the uncertainty in the sample locations (see Naimi et al. 2011, for more information).

**Value**

RasterVariogram

**Author(s)**

Babak Naimi <naimi.b@gmail.com>

<http://r-gis.net>

**References**

Naimi, B., Skidmore, A.K., Groen, T.A., Hamm, N.A.S. 2011. Spatial autocorrelation in predictors reduces the impact of positional uncertainty in occurrence data on species distribution modelling, *Journal of biogeography*. 38: 1497-1509.

Naimi, B., Hamm, N.A.S., Groen, T.A., Skidmore, A.K., and Toxopeus, A.G. 2014. Where is positional uncertainty a problem for species distribution modelling?, *Ecography* 37 (2): 191-203.

**Examples**

```
## Not run:
file <- system.file("external/spain.grd", package="usdm")

r <- brick(file) # reading a RasterBrick object including 10 raster layers in Spain

r

plot(r[[1]]) # plot the first RasterLayer in r

v1 <- Variogram(r[[1]]) # compute the sample variogram for the first layer in r

v2 <- Variogram(r[[1]],lag=25000,cutoff=100000) # specify the lag and cutoff parameters

## End(Not run)
```

vif

*Variance Inflation Factor and test for multicollinearity***Description**

Calculates variance inflation factor (VIF) for a set of variables and exclude the highly correlated variables from the set through a stepwise procedure. This method can be used to deal with multicollinearity problems when you fit statistical models

**Usage**

```
vif(x, ...)
vifcor(x, th=0.9, ...)
vifstep(x, th=10, ...)
```

**Arguments**

x	explanatory variables (predictors), defined as a raster object (RasterStack or RasterBrick), or as a matrix, or as a data.frame.
th	a number specifying the correlation threshold for vifcor and VIF threshold for vifstep (see details).
...	additional arguments. see details.

**Details**

VIF can be used to detect collinearity (Strong correlation between two or more predictor variables). Collinearity causes instability in parameter estimation in regression-type models. The VIF is based on the square of the multiple correlation coefficient resulting from regressing a predictor variable against all other predictor variables. If a variable has a strong linear relationship with at least one other variables, the correlation coefficient would be close to 1, and VIF for that variable would be

large. A VIF greater than 10 is a signal that the model has a collinearity problem. `vif` function calculates this statistic for all variables in `x`. `vifcor` and `vifstep` uses two different strategy to exclude highly collinear variable through a stepwise procedure. `vifcor`, first find a pair of variables which has the maximum linear correlation (greater than `th`), and exclude one of them which has greater VIF. The procedure is repeated until no variable with a high correlation coefficient (greater than `threshold`) with other variables remains. `vifstep` calculate VIF for all variables, exclude one with highest VIF (greater than `threshold`), repeat the procedure until no variables with VIF greater than `th` remains.

additional arguments:

`maxobservations` a number (default=5000) specifying the maximum number of observations should be contributed in calculation of VIF. When the number of observations (cells in raster or rows in `data.frame/matrix`) is greater than `maxobservations`, then a random sample with a size of `maxobservations` is drawn to keep the calculation efficient.

### Value

an object of class VIF

### Author(s)

Babak Naimi <naimi.b@gmail.com>

<http://r-gis.net>

### References

Chatterjee, S. and Hadi, A. S. 2006. Regression analysis by example. John Wiley and Sons.;  
 Dormann, C. F. et al. 2012. Collinearity: A review of methods to Deal with it and a simulation study evaluating their performance. *Ecography* 35: 001-020.;

---

If you used this method, please cite the following article for which this package is developed:

Naimi, B., Hamm, N.A.S., Groen, T.A., Skidmore, A.K., and Toxopeus, A.G. 2014. Where is positional uncertainty a problem for species distribution modelling?, *Ecography* 37 (2): 191-203.

### See Also

[exclude](#)

### Examples

```
## Not run:
file <- system.file("external/spain.grd", package="usdm")

r <- brick(file) # reading a RasterBrick object including 10 raster layers in Spain

r

vif(r) # calculates vif for the variables in r
```

```
v1 <- vifcor(r, th=0.9) # identify collinear variables that should be excluded
v1
v2 <- vifstep(r, th=10) # identify collinear variables that should be excluded
v2
## End(Not run)
```

---

VIF-class

*VIF class*

---

### Description

An object of the VIF class contains information about collinearity in relevant variables. The object can be created with the following functions: [vifcor](#) and [vifstep](#).

### Slots

Slots for VIF object

**variables:** Character

**excluded:** character

**corMatrix:** a correlation matrix

**results:** data.frame including VIF values for the remained (not excluded) variables

### Author(s)

Babak Naimi <naimi.b@gmail.com>

<http://r-gis.net>

### Examples

```
showClass("VIF")
```



# Index

## \*Topic **classes**

RasterVariogram-class, 9  
speciesLISA-class, 12  
VIF-class, 16

## \*Topic **methods**

plot.RasterVariogram, 6  
plot.speciesLISA, 7

## \*Topic **package**

usdm-package, 2

## \*Topic **spatial**

exclude, 2  
lisa, 4  
plot.RasterVariogram, 6  
plot.speciesLISA, 7  
RasterVariogram-class, 9  
speciesLisa, 10  
speciesLISA-class, 12  
usdm-package, 2  
Variogram, 12  
vif, 14  
VIF-class, 16

exclude, 2, 15

exclude,data.frame,missing-method  
(exclude), 2

exclude,data.frame,VIF-method  
(exclude), 2

exclude,matrix,missing-method  
(exclude), 2

exclude,matrix,VIF-method (exclude), 2

exclude,RasterBrick,missing-method  
(exclude), 2

exclude,RasterBrick,VIF-method  
(exclude), 2

exclude,RasterStack,missing-method  
(exclude), 2

exclude,RasterStack,VIF-method  
(exclude), 2

lisa, 4, 8, 11

lisa,Raster,missing-method (lisa), 4

lisa,Raster,SpatialPoints-method  
(lisa), 4

lisa,Raster,SpatialPointsDataFrame-method  
(lisa), 4

plot,RasterVariogram-method  
(plot.RasterVariogram), 6

plot,speciesLISA,missing-method  
(plot.speciesLISA), 7

plot,speciesLISA,SpatialPolygons-method  
(plot.speciesLISA), 7

plot,speciesLISA,SpatialPolygonsDataFrame-method  
(plot.speciesLISA), 7

plot.RasterVariogram, 6

plot.speciesLISA, 7

print,speciesLISA-method  
(speciesLISA-class), 12

RasterVariogram-class, 9

show,RasterVariogram-method  
(RasterVariogram-class), 9

show,VIF-method (VIF-class), 16

speciesLisa, 8, 10, 12

speciesLisa,Raster,SpatialPoints-method  
(speciesLisa), 10

speciesLisa,Raster,SpatialPointsDataFrame-method  
(speciesLisa), 10

speciesLISA-class, 12

usdm (usdm-package), 2

usdm-package, 2

Variogram, 6, 7, 9, 12

Variogram,RasterLayer-method  
(Variogram), 12

vif, 3, 14

vif,data.frame-method (vif), 14

vif,matrix-method (vif), 14

vif,RasterStackBrick-method (vif), 14

VIF-class, [16](#)  
vifcor, [16](#)  
vifcor(vif), [14](#)  
vifcor,data.frame-method(vif), [14](#)  
vifcor,matrix-method(vif), [14](#)  
vifcor,RasterStackBrick-method(vif), [14](#)  
vifstep, [16](#)  
vifstep(vif), [14](#)  
vifstep,data.frame-method(vif), [14](#)  
vifstep,matrix-method(vif), [14](#)  
vifstep,RasterStackBrick-method(vif),  
[14](#)