

Package ‘SpatialFloor’

April 26, 2018

Type Package

Title Spatial Floor Simulation (Isotropic)

Version 1.1.0

Date 2018-04-20

Author Le Zheng <le.zheng@advantaseeds.com>, Jose Alfaro <JFA170001@utdallas.edu>, Martin Grondona <martin.grondona@advantaseeds.com>, Maria Arrieta <maria.arrieta@advantaseeds.com>

Maintainer Le Zheng <le.zheng@advantaseeds.com>

License GPL (>= 2)

Depends R (>= 3.0.0), fields, reshape2

Imports stats, base, taRifx, blocksdesign

Description Spatial floor simulation with exponential/Gaussian variance-covariance function (isotropic), with specification of distance function, nugget, sill, range. The methodology follows Nole A.C. Cressie (2015) <doi:10.1002/9781119115151>. The original release is 2017-08-29.

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-04-26 13:45:32 UTC

R topics documented:

check_layout_generation	2
density_check	3
diagonal_check_by_density	3
diagSet_check_by_density	4
diag_extract	4
equal_check_by_plot	5
equal_space_check_by_density	5
iso_simulation_by_heritability	6
iso_simulation_p_rep	7

iso_spatial_cov	8
iso_spatial_exp	9
iso_spatial_floor	10
iso_spatial_gau	10
p_rep_check_by_density	11
random_check_by_density	12
spatial_dist	12

Index	14
--------------	-----------

check_layout_generation

This is a wrapper function of check layout generation

Description

The function takes grid input, and generates the corresponding check layout

Usage

```
check_layout_generation(n.row, n.col, by.density, gen.module,
  p_rep_check = 0.05)
```

Arguments

n.row	The row of the grid layout
n.col	The column of the grid layout
by.density	The density of the check in the layout. The parameter can be a vector
gen.module	The pattern module of check layout, 'diagonal', 'diagSet', 'random', 'equal_space', 'p_rep'. The diagSet provides more homogeneized design than the diagonal at density from 0.01 to 0.35. The parameter can be a vector
p_rep_check	The check density in the p_rep design, default 0.05

Examples

```
nr = nc = 5; den = c(0.1, 0.2); gen.module=c('diagSet', 'random')
check_layout_generation(nr, nc, den, gen.module)
```

density_check	<i>Internal function for density check of check layout</i>
---------------	------------------------------------------------------------

Description

This is an internal function

Usage

```
density_check(layout.matrix, by.density)
```

Arguments

layout.matrix	The layout matrix of designed grid
by.density	The check density in the layout matrix

diagonal_check_by_density	<i>Generate a spatial layout of checks by diagonal design</i>
---------------------------	---------------------------------------------------------------

Description

This function generates a layout grid with checks on diagonal

Usage

```
diagonal_check_by_density(n.row, n.col, by.density, spread.scale = 50 *
  max(n.row, n.col))
```

Arguments

n.row	The row of the rectangular grid
n.col	The column of the rectangular grid
by.density	The density of the check
spread.scale	The spreading of the check across diagonal, related to the scale parameter in Cauchy distribution, default=50*max(n.row, n.col)

Examples

```
#Create a 30 by 30 layout with 0.2 density
tmp = diagonal_check_by_density(30, 30, 0.2)
fields::quilt.plot(tmp, nx=30, ny=30)
```

diagSet_check_by_density

Generate a spatial layout of checks by diagonal design (settle for density = .1 - .35)

Description

This function generates a layout grid with checks on diagonal (pre-designed)

Usage

```
diagSet_check_by_density(n.row, n.col, by.density)
```

Arguments

n.row	The row of the rectangular grid
n.col	The column of the rectangular grid
by.density	The density of the checks, range from 0.01 to 0.35

Examples

```
nr=10; nc=20; den=0.2  
diagSet_check_by_density(n.row=nr, n.col=nc, by.density=den)
```

diag_extract

Extract the designed diagonal pattern from the system data file (Internal functions)

Description

Functions of extracting the pre-designed diagonal pattern (density from 0.01 up to 0.35)

Usage

```
diag_extract()
```

equal_check_by_plot *Generate a spatial layout of checks by every # of plots*

Description

This function generate a check layout in a rectangular grid, The check is distributed by every # of plots

Usage

```
equal_check_by_plot(n.row, n.col, by.plot)
```

Arguments

n.row	The row of a rectangular grid
n.col	The column of a rectangular grid
by.plot	The distance between checks (row-wise and col-wise) in the unit of plots

Value

A dataframe will return

Examples

```
# To produce a 4 by 5 field with check distributed by every 2 plots
nr = 4; nc = 5
example.tmp = equal_check_by_plot(nr, nc, 2)
fields::quilt.plot(example.tmp, nx=nr, ny=nc)
```

equal_space_check_by_density
Generate a spatial layout of checks by density

Description

This function generate a check layout in a rectangular grid, The check is evenly distributed by the given density

Usage

```
equal_space_check_by_density(n.row, n.col, by.density)
```

Arguments

n.row	The row of a rectangular grid
n.col	The column of a rectangular grid
by.density	The density of the check

Value

A dataframe will return

Examples

```
# To produce a 4 by 5 field with check distributed by density=0.2
nr = 4; nc = 5
example.tmp = equal_space_check_by_density(nr, nc, .2)
fields::quilt.plot(example.tmp, nx=nr, ny=nc)
```

iso_simulation_by_heritability

A simulation function for isotropic spatial data

Description

The function simulates spatial data by given parameters

Usage

```
iso_simulation_by_heritability(n.row, n.col, lon.lat = FALSE, mile = FALSE,
  density.choice, density.layout = c("diagSet", "random", "equal_space",
  "p_rep"), h2, sigma_env, sigma_variety = (sigma_env * h2)/(1 - h2),
  mu_variety, mu_check = mu_variety + 1.68 * sqrt(sigma_variety),
  cov_fun = "exp", ranges = sqrt(2), simulation = 3, nugget = 0,
  mu_floor = 0, p_rep_check = 0.05)
```

Arguments

n.row	(Mandatory) The row of the field grid
n.col	(Mandatory) The column of the fields grid
lon.lat	(Optional) The distance calculation is based on earth distance (if TRUE), default is FALSE
mile	(Optional) The distance calculation is based on milage (if TRUE), default is FALSE
density.choice	(Mandatory) The density of the checks, a vector (range from 0 to 1)
density.layout	(Optional) The layout pattern of density, default are: 'diagonal', 'diagSet', 'random', 'equal_space', 'p_rep'. The diagSet is recommended over the diagonal at density from 0.01 to 0.35.

h2	(Mandatory) The heritability of the simulated data, $h2 = \text{sigma_variety} / (\text{sigma_variety} + \text{sigma_env})$
sigma_env	(Mandatory) The variance of environment, (equivalent of sill)
sigma_variety	(Optional) The variance of entry, default $\text{sigma_variety} = (\text{sigma_env} * h2) / (1 - h2)$
mu_variety	(Mandatory) The mean of the variety
mu_check	(Optional) The mean value of checks, default $\text{mu_check} = 1.68 * \text{sqrt}(\text{sigma_variety})$
cov_fun	(Optional) The var-cov function ('exp' or 'gau'), default is 'exp', exponential
ranges	(Optional) The range parameter of spatial distribution, default $\text{sqrt}(2)$
simulation	(Optional) The number of simulated data returned, default 3
nugget	(Optional) The nugget parameter in spatial simulation, default 0
mu_floor	(Optional) The mean of the spatial floor, default 0
p_rep_check	The density of checks in the p rep design, default 0.05

Examples

```
temp.1 = iso_simulation_by_heritability(20, 20, density.choice=c(0.05)
, density.layout=c('diagSet', 'random')
, h2=0.5, sigma_env=100, mu_variety=300)
temp.2 = iso_simulation_by_heritability(20, 20, density.choice=c(0.05)
, density.layout=c('diagSet', 'random')
, h2=0.5, sigma_env=100, mu_variety=300)
```

iso_simulation_p_rep *A simulation function for isotropic spatial data wrt/ p-rep data*

Description

The function simulates spatial data by given parameters

Usage

```
iso_simulation_p_rep(n.row, n.col, lon.lat = FALSE, mile = FALSE,
density.choice, density.layout = c("diagSet", "random", "equal_space"), h2,
sigma_env, sigma_variety = (sigma_env * h2)/(1 - h2), mu_variety,
cov_fun = "exp", ranges = sqrt(2), simulation = 3, nugget = 0,
mu_floor = 0)
```

Arguments

n.row	(Mandatory) The row of the field grid
n.col	(Mandatory) The column of the fields grid
lon.lat	(Optional) The distance calculation is based on earth distance (if TRUE), default is FALSE

mile	(Optional) The distance calculation is based on milage (if TRUE), default is FALSE
density.choice	(Mandatory) The density of the replicated entries (range from 0 to 1, if 'diagonal'; from 0 to 0.34, if 'diagSet'), EVEN number ONLY
density.layout	(Optional) The layout pattern of density, default are: 'diagonal', 'diagSet', 'random', 'equal_space'. The diagSet is recommended over the diagonal at density from 0.01 to 0.35.
h2	(Mandatory) The heritability of the simulated data, $h2 = \text{sigma_variety} / (\text{sigma_variety} + \text{sigma_env})$
sigma_env	(Mandatory) The variance of environment, (equilievment of sill)
sigma_variety	(Optional) The variance of entry, default $\text{sigma_variety} = (\text{sigma_env} * h2) / (1 - h2)$
mu_variety	(Mandatory) The mean of the variety
cov_fun	(Optional) The var-cov function ('exp' or 'gau'), default is 'exp', exponential
ranges	(Optional) The range parameter of spatial distribution, default sqrt(2)
simulation	(Optional) The number of simulated data returned, default 3
nugget	(Optional) The nugget parameter in spatial simulation, default 0
mu_floor	(Optional) The mean of the spatial floor, default 0

Examples

```
temp.1 = iso_simulation_p_rep(10, 10, density.choice=c(0.04,0.2)
, h2=0.5, sigma_env=100, mu_variety=300)
temp.2 = iso_simulation_p_rep(10, 10, density.choice=c(0.04,0.2)
, h2=0.5, sigma_env=100, mu_variety=300)
```

iso_spatial_cov *Isotropic spatial var-cov wrapper*

Description

This function load grid floor and generate the var-cov matrix

Usage

```
iso_spatial_cov(D, cov.var = "exp", nugget = 0, sill, ranges)
```

Arguments

D	The distance matrix
cov.var	the var-cov function of producing var-cov matrix. 'exp' is exponential, 'gau' is Gaussian
nugget	the nugget value of the var-cov function
sill	the sill(variance, sigma^2) of the var-cov function
ranges	the range value of var-cov function

Value

A matrix of var-cov matrix

Examples

```
D = fields::rdist(expand.grid(1:5, 1:5))
iso_spatial_cov(D, sill=3, ranges=1)
```

iso_spatial_exp	<i>Isotropic spatial simulation by exponential function</i>
-----------------	-------------------------------------------------------------

Description

This function load grid floor and generate the var-cov matrix via exponential function

Usage

```
iso_spatial_exp(dist.matrix, nugget, sill, ranges)
```

Arguments

dist.matrix	the a distance matrix generate from spatial_dist function
nugget	the nugget value of the var-cov function
sill	the sill(variance, σ^2) of the var-cov function
ranges	the range value of var-cov function

Value

A matrix of var-cov matrix

Examples

```
D = spatial_dist(expand.grid(1:5, 1:5))
iso_spatial_exp(D, 0, 3, 1)
```

iso_spatial_floor *Isotropic spatial simulation wrapper*

Description

This function load grid floor and generate the simulated floor

Usage

```
iso_spatial_floor(D, cov.var = "exp", nugget = 0, sill, ranges, mu = 0)
```

Arguments

D	The distance matrix
cov.var	the var-cov function of producing var-cov matrix. 'exp' is exponential, 'gau' is Gaussian
nugget	the nugget value of the var-cov function
sill	the sill(variance, σ^2) of the var-cov function
ranges	the range value of var-cov function
mu	the mean value of the simulated spatial floor

Value

A vector with the same length of grid floor

Examples

```
D = spatial_dist(expand.grid(1:5, 1:5))
iso_spatial_floor(D, sill=50, ranges=2)
```

iso_spatial_gau *Isotropic spatial simulation by Gaussian function*

Description

This function load grid floor and generate the var-cov matrix via Gaussian function

Usage

```
iso_spatial_gau(dist.data, nugget, sill, ranges)
```

Arguments

dist.data	the a distance matrix generate from spatial_dist function
nugget	the nugget value of the var-cov function
sill	the sill(variance, σ^2) of the var-cov function
ranges	the range value of var-cov function

Value

A matrix of var-cov matrix

Examples

```
D = spatial_dist(expand.grid(1:5, 1:5))
iso_spatial_gau(D, 0, 3, 1)
```

p_rep_check_by_density

Generate a spatial layout of checks by density: p rep design with checks

Description

This function generate a check layout in a rectangular grid. the check and partial replicated entries are spread by blocksdesign library

Usage

```
p_rep_check_by_density(n.row, n.col, check.density = 0.05, by.density)
```

Arguments

n.row	The row of a rectangular grid
n.col	The column of a rectangular grid
check.density	The density of the check
by.density	The total density of check and p_rep, MUST be a even decimal number after subtract the check.density

Value

A dataframe will return: row.var=row, col.var=col, check=0, single rep; 1, check; else, partial replicated entries

Examples

```
# To produce a 20 by 20 field with check distributed by density=0.35
nr = 20; nc = 20
example.tmp = p_rep_check_by_density(nr, nc, by.density=.35)
fields::quilt.plot(example.tmp, nx=nr, ny=nc)
```

random_check_by_density

Generate a random spatial layout of check

Description

This function generate a random spatial layout of checks

Usage

```
random_check_by_density(n.row, n.col, by.density)
```

Arguments

n.row	The row of the rectangular grid
n.col	The column of the rectangular grid
by.density	The density of checks

Examples

```
nr=nc=5
tmp = random_check_by_density(nr, nc, .1)
fields::quilt.plot(tmp, nx=nr, ny=nc)
```

spatial_dist

Distance matrix calculation

Description

This function load grid floor and generate the distance matrix

Usage

```
spatial_dist(M, lon.lat = F, mile = F)
```

Arguments

M	A data.frame contains grid coordinates
lon.lat	If TRUE, the function will use longitude and latitude for calculating geographic distance
mile	If TRUE, the function will yeild distance in mileages. If FALSE, the function will yeild distance in kilometers

Value

A matrix of distance matrix

Examples

```
spatial_dist(expand.grid(1:4, 1:4))
```

Index

check_layout_generation, [2](#)

density_check, [3](#)
diag_extract, [4](#)
diagonal_check_by_density, [3](#)
diagSet_check_by_density, [4](#)

equal_check_by_plot, [5](#)
equal_space_check_by_density, [5](#)

iso_simulation_by_heritability, [6](#)
iso_simulation_p_rep, [7](#)
iso_spatial_cov, [8](#)
iso_spatial_exp, [9](#)
iso_spatial_floor, [10](#)
iso_spatial_gau, [10](#)

p_rep_check_by_density, [11](#)

random_check_by_density, [12](#)

spatial_dist, [12](#)