Package: simsam (via r-universe)

May 11, 2026

Title Simulating and Sampling Spatial Data

Version 0.2.3

Description Provides tools to simulate and sample spatial data. Additional tools for creating various spatial proxies are also included.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports gstat, terra, sf

Suggests tinytest

Config/Needs/website rmarkdown

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

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

Date/Publication 2026-05-11 06:23:30 UTC

RemoteUrl https://github.com/nowosad/simsam

RemoteRef HEAD

RemoteSha a07f1346939d92d9203f719512ec38090849288b

Contents

`blend_rasters` *Blend rasters based on a provided formula*

Description

Blend rasters based on a provided formula

Usage

```
blend_rasters(x, formula, ...)
```

Arguments

`x`	A `SpatRaster` object with one or more layers
`formula`	A formula specifying how to combine the rasters
`...`	Additional `SpatRaster` objects to blend

Value

A `SpatRaster` object with the combined layers

Examples

```
rast_grid = terra::rast(ncols = 300, nrows = 100,
  xmin = 0, xmax = 300, ymin = 0, ymax = 100)

sf1 = sim_covariates(
  rast_grid,
  n = 4,
  method = simulate_gaussian(range = 25)
)

g1 = blend_rasters(sf1, ~ cov1 + cov4)
g2 = blend_rasters(g1, ~ outcome^2 + (cov2 + 2), sf1)

terra::plot(g1)
terra::plot(g2)
```

make_proxy *Generate spatial proxies*

Description

The make_proxy() function generates spatial proxies for a given raster object. It generates spatial proxies such as:

- coordinates
- Euclidean Distance Fields (EDF)
- Oblique Geographic Coordinates (OGC)

Usage

```
make_proxy(rast_grid, method = proxy_coordinates())

add_proxy(rast_grid, method = proxy_coordinates())
```

Arguments

rast_grid	A SpatRaster
method	A proxy method created by proxy_*() functions

Value

A SpatRaster object

SpatRaster

References

Behrens, T., Schmidt, K., Viscarra Rossel, R. A., Gries, P., Scholten, T., & MacMillan, R. A. (2018). Spatial modelling with Euclidean distance fields and machine learning. European journal of soil science, 69(5), 757-770.

Møller, A. B., Beucher, A. M., Pouladi, N., & Greve, M. H. (2020). Oblique geographic coordinates as covariates for digital soil mapping. Soil, 6(2), 269-289.

Examples

```
rast_grid = terra::rast(
  ncols = 300, nrows = 100,
  xmin = 0, xmax = 300,
  ymin = 0, ymax = 100
)

proxy_coords = make_proxy(rast_grid, proxy_coordinates())
proxy_edf = make_proxy(rast_grid, proxy_edf())
proxy_ogc = make_proxy(rast_grid, proxy_ogc(n = 5))
```

```
terra::plot(proxy_coords)
terra::plot(proxy_edf)
terra::plot(proxy_ogc)
```

`proxy_coordinates` *Coordinate proxy factory*

Description

Returns a function that generates coordinate layers (X, Y) for a raster.

Usage

```
proxy_coordinates()
```

Value

A function that accepts `rast_grid` and returns a `SpatRaster`

Examples

```
rast_grid = terra::rast(
  ncols = 300, nrows = 100,
  xmin = 0, xmax = 300,
  ymin = 0, ymax = 100
)

make_proxy(rast_grid, proxy_coordinates())
```

`proxy_edf` *Euclidean Distance Field proxy factory*

Description

Returns a function that generates Euclidean Distance Fields from specified locations (default: four corners and center of the raster extent).

Usage

```
proxy_edf(coords = NULL)
```

Arguments

`coords`	Optional coordinates. Can be a matrix, data frame, `sf`, or `SpatVector` with x and y columns. If `NULL`, uses default locations: four corners and center of the raster extent.

Value

A function that accepts `rast_grid` and returns a `SpatRaster`

Examples

```
rast_grid = terra::rast(
  ncols = 300, nrows = 100,
  xmin = 0, xmax = 300,
  ymin = 0, ymax = 100
)

# Default: four corners and center
make_proxy(rast_grid, proxy_edf())

# Custom coordinates
custom_coords = matrix(c(50, 50, 150, 150), ncol = 2, byrow = TRUE)
rownames(custom_coords) = c("point1", "point2")
make_proxy(rast_grid, proxy_edf(coords = custom_coords))
```

---

proxy_ogc *Oblique Geographic Coordinates proxy factory*

Description

Returns a function that generates Oblique Geographic Coordinates (OGC) as covariates.

Usage

```
proxy_ogc(n = 5)
```

Arguments

`n`	Number of angles (default: 5)

Value

A function that accepts `rast_grid` and returns a `SpatRaster`

References

Møller, A. B., Beuchert, A. M., Pouladi, N., & Greve, M. H. (2020). Oblique geographic coordinates as covariates for digital soil mapping. Soil, 6(2), 269-289.

Examples

```
rast_grid = terra::rast(
  ncols = 300, nrows = 100,
  xmin = 0, xmax = 300,
  ymin = 0, ymax = 100
)

make_proxy(rast_grid, proxy_ogc(n = 5))
```

---

sam_field *Sample spatial field*

Description

Create a sample of a spatial field.

Usage

```
sam_field(x, size, method = sample_random())
```

Arguments

x	A raster object (SpatRaster).
size	Number of samples to create.
method	Sampling method (function created by sample_random(), sample_jittered(), or sample_clustered()).

Value

An sf object with sampled points

Examples

```
rast_grid = terra::rast(
  ncols = 300, nrows = 100,
  xmin = 0, xmax = 300,
  ymin = 0, ymax = 100
)

sam_field(
  rast_grid,
  100,
  method = sample_jittered(amount = 5)
)
```

sample_clustered *Clustered sampling method*

Description

Returns a function that performs clustered sampling on a spatial field. Cluster centers are randomly distributed, then samples are drawn within a buffer radius around each cluster.

Usage

```
sample_clustered(nclusters, radius, ...)
```

Arguments

nclusters	Number of clusters
radius	Cluster radius (in map units)
...	Additional arguments passed to terra::spatSample()

Value

A function that accepts x (SpatRaster) and size and returns an sf object

Examples

```
rast_grid = terra::rast(
  ncols = 300, nrows = 100,
  xmin = 0, xmax = 300,
  ymin = 0, ymax = 100
)

sam_field(rast_grid, 200, method = sample_clustered(nclusters = 5, radius = 10))
```

sample_jittered *Jittered sampling method*

Description

Returns a function that performs jittered sampling on a spatial field. Regular points are generated first, then jittered by a random amount.

Usage

```
sample_jittered(amount, ...)
```

Arguments

amount	Jitter amount in map units
...	Additional arguments passed to terra::spatSample()

Value

A function that accepts x (SpatRaster) and size and returns an sf object

Examples

```
rast_grid = terra::rast(
  ncols = 300, nrows = 100,
  xmin = 0, xmax = 300,
  ymin = 0, ymax = 100
)

sam_field(rast_grid, 100, method = sample_jittered(amount = 5))
```

---

sample_random *Random sampling method*

Description

Returns a function that performs random sampling on a spatial field.

Usage

```
sample_random(...)
```

Arguments

...	Additional arguments passed to terra::spatSample()

Value

A function that accepts x (SpatRaster) and size and returns an sf object

Examples

```
rast_grid = terra::rast(
  ncols = 300, nrows = 100,
  xmin = 0, xmax = 300,
  ymin = 0, ymax = 100
)

sam_field(rast_grid, 100, method = sample_random())
```

`sim_covariates` *Simulate a raster stack of covariates*

Description

Generic function for simulating spatial covariates using a specified method.

Usage

```
sim_covariates(rast_grid, n = 6, method)
```

Arguments

`rast_grid`	A `SpatRaster` object with the desired dimensions
`n`	The number of covariates to simulate (must be >= 1)
`method`	A simulation **factory function** —- call the function with parameters to create the method (remember to not pass the function itself). For example: `method = simulate_gaussian(range = 25)`, not `method = simulate_gaussian`. See `simulate_gaussian()` and `simulate_random()` for examples.

Value

A `SpatRaster` with `n` layers named `cov1`, `cov2`, ..., `covN`

Examples

```
rast_grid = terra::rast(ncols = 300, nrows = 100,
  xmin = 0, xmax = 300, ymin = 0, ymax = 100)
# Gaussian simulation
sf1 = sim_covariates(rast_grid, n = 4, method = simulate_gaussian(range = 25))
sf2 = sim_covariates(rast_grid, n = 4, method = simulate_gaussian(range = 50, model = "Exp"))
# Reusable simulation engine
gauss = simulate_gaussian(range = 25)
sf3 = sim_covariates(rast_grid, n = 4, method = gauss)
# Random simulation
sf4 = sim_covariates(rast_grid, n = 4, method = simulate_random())
terra::plot(sf1)
terra::plot(sf2)
terra::plot(sf3)
terra::plot(sf4)
```

`simulate_gaussian` *Simulate Gaussian random fields*

Description

Returns a function that simulates spatially correlated Gaussian random fields using a variogram model.

Usage

```
simulate_gaussian(
  vgm = NULL,
  psill = 1,
  model = "Sph",
  range = NULL,
  nugget = 0,
  beta = 0,
  nmax = 30,
  indicators = FALSE,
  seed = NULL
)
```

Arguments

`vgm`	A variogram model object (`variogramModel` or `gstatVariogramModel`). If `NULL`, one is built from `model`, `psill`, `range`, and `nugget`.
`psill`	Partial sill (default: 1). Ignored if `vgm` is provided.
`model`	Variogram model type (e.g., `"Sph"`, `"Exp"`, `"Gau"`, `"Nug"`). Ignored if `vgm` is provided.
`range`	Spatial range parameter. **Required** if `vgm` is `NULL`.
`nugget`	Nugget effect (default: 0). Ignored if `vgm` is provided.
`beta`	Mean of the Gaussian field (default: 0). Passed to `gstat::gstat()`.
`nmax`	The number of nearest observations used for kriging simulation (default: 30). Passed to `gstat::gstat()`.
`indicators`	Indicator thresholds (default: `FALSE`). Passed to `stats::predict()`. Use `TRUE` for default thresholds or a numeric vector for custom thresholds.
`seed`	Optional random seed for reproducibility. Set this to ensure that the same random field is generated every time the returned function is called.

Value

A function that accepts `rast_grid` and `n` and returns a `SpatRaster`

Examples

```
rast_grid = terra::rast(ncols = 300, nrows = 100,
  xmin = 0, xmax = 300, ymin = 0, ymax = 100)
# Direct usage
sim_fn = simulate_gaussian(range = 25)
sf1 = sim_covariates(rast_grid, n = 4, method = sim_fn)
# With custom variogram
vgm = gstat::vgm(model = "Exp", psill = 1, range = 10)
sim_fn2 = simulate_gaussian(vgm = vgm)
sf2 = sim_covariates(rast_grid, n = 4, method = sim_fn2)
terra::plot(sf1)
terra::plot(sf2)
```

---

simulate_random *Simulate uncorrelated random fields*

Description

Returns a function that simulates uncorrelated random fields (white noise).

Usage

```
simulate_random(mean = 0, sd = 1, seed = NULL)
```

Arguments

mean	Mean of the random values (default: 0)
sd	Standard deviation of the random values (default: 1)
seed	Optional random seed for reproducibility

Value

A function that accepts `rast_grid` and `n` and returns a `SpatRaster`

Examples

```
rast_grid = terra::rast(ncols = 300, nrows = 100,
  xmin = 0, xmax = 300, ymin = 0, ymax = 100)
sim_fn = simulate_random()
sf1 = sim_covariates(rast_grid, n = 4, method = sim_fn)
terra::plot(sf1)
```

Index