Package ‘grainchanger’

February 1, 2021

Title Moving-Window and Direct Data Aggregation

Version 0.3.2

Description Data aggregation via moving window or direct methods. Aggregate a fine-resolution raster to a grid. The moving window method smooths the surface using a specified function within a moving window of a specified size and shape prior to aggregation. The direct method simply aggregates to the grid using the specified function.

Depends R (>= 3.3)

License GPL-3

Encoding UTF-8

LazyData true

Imports raster, sf, furrr, checkmate, methods, usethis

Suggests dplyr (>= 0.8-0), testthat, knitr, rmarkdown, covr, ggplot2, landscapetools, rgdal

RoxygenNote 7.1.1

Language en-GB

VignetteBuilder knitr

URL https://docs.ropensci.org/grainchanger/, https://github.com/ropensci/grainchanger

BugReports https://github.com/ropensci/grainchanger/issues

NeedsCompilation no

Author Laura Graham [aut, cre] (<https://orcid.org/0000-0002-3611-7281>), Felix Eigenbrod [ctb] (Input on initial conceptual development), Marco Sciaini [ctb] (Input on package development and structure)

Maintainer Laura Graham <LauraJaneEGraham@gmail.com>

Repository CRAN

Date/Publication 2021-02-01 15:20:08 UTC
R topics documented:

- cat_ls .................................................. 2
- cont_ls .................................................. 3
- create_torus .......................................... 3
- diversity-metrics ...................................... 4
- g_sf ...................................................... 5
- mean ..................................................... 6
- nomove_agg ............................................ 7
- poly_sf .................................................. 8
- prop ...................................................... 8
- var_range .............................................. 9
- winmove ................................................ 10
- winmove-class ........................................ 11
- winmove_agg .......................................... 12

Index 15

---

cat_ls  

Example categorical raster (fine_dat)

Description

An example map to show functionality on categorical surfaces.

Usage

cat_ls

Format

A raster layer object.

Details

Generated with nlm_mpd() from NLMR and classified with util_classify() from landscapetools.

Source


**cont_ls**  
*Example continuous raster (fine_dat)*

---

**Description**

An example map to show functionality on continuous surfaces.

**Usage**

```r
cont_ls
```

**Format**

A raster layer object.

**Details**

Generated with `nlm_mpd()` from NLMR.

**Source**


---

**create_torus**  
*Pad a raster by a specified radius*

---

**Description**

This function pads a raster by a specified number of cells, creating the effect of a torus. This function is intended for use on simulated landscapes, in order to avoid edge effects.

**Usage**

```r
create_torus(dat, dpad)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat</td>
<td>The raster dataset to pad</td>
</tr>
<tr>
<td>dpad</td>
<td>The amount by which to pad the raster (in the same units as the raster)</td>
</tr>
</tbody>
</table>
Details

A torus is an infinite surface where the top joins the bottom, and the left side meets the right side. See https://en.wikipedia.org/wiki/Torus for a full mathematical description.

In this function, the torus effect is achieved by adding the specified number of rows of the top of the raster to the bottom (and vice versa) and the specified number of rows of the right of the raster to the left (and vice versa)

Value

raster. Original raster padded by r cells with torus effect (see Details)

Examples

data(cat_ls)
d <- create_torus(dat = cat_ls, dpad = 5)

Description

A range of functions to calculate well known landcover diversity metrics

Usage

## S3 method for class 'winmove'
shdi(x, lc_class, d, type, ...)

## S3 method for class 'numeric'
shdi(x, lc_class, ...)

## S3 method for class 'winmove'
shei(x, lc_class, d, type, ...)

## S3 method for class 'numeric'
shei(x, lc_class, ...)

Arguments

x numeric, winmove. The data over which to calculate the diversity metrics

lc_class numeric. The class values to include in the diversity metric calculation

d numeric. If type=circle, the radius of the circle (in units of the CRS). If type=rectangle the dimension of the rectangle (one or two numbers)

type character. The shape of the moving window

... further arguments passed to or from other methods
Details

Currently provided diversity metrics are Shannon diversity and Shannon evenness. Open a new issue (https://github.com/laurajanegraham/grainchanger/issues) to request additional diversity metrics.

Value

If `class(x) == "winmove"`, a smoothed raster with the diversity metric calculated within the specified moving window

If `class(x) == "numeric"`, a single value representing the diversity metric in `x`

References


Examples

```r
# load required data
data(cat_ls)

# convert data to object of class winmove
cat_ls <- new("winmove", cat_ls)

# calculate Shannon diversity in a rectangular window of dimension 5
d <- shdi(cat_ls, d = 5, type = "rectangle", lc_class = 1:4)

# convert data to object of class numeric
cat_ls <- raster::values(cat_ls)

# calculate Shannon evenness
d <- shei(cat_ls, lc_class = 1:4)
```

---

**g_sf**  
*Example grid (coarse.dat)*

Description

An example grid to show functionality when aggregating using an sf object.

Usage

```
g_sf
```

Format

An sf object.
Details

Generated with nlm_mpd() and converted to sf.

Source


mean

Arithmetic mean

Description

An extension to mean for objects of class winmove

Usage

mean(x, ...)

## S3 method for class 'winmove'
mean(x, d, type, ...)

Arguments

x  RasterLayer. The data over which to calculate the mean value within a moving window
...
  further arguments passed to or from other methods
d  numeric. If type=circle, the radius of the circle (in units of the CRS). If type=rectangle the dimension of the rectangle (one or two numbers)
type  character. The shape of the moving window

Value

RasterLayer. A smoothed raster with the mean calculated within the specified moving window

Examples

# load required data
data(cont_ls)

# convert data to object of class winmove
cont_ls <- new("winmove", cont_ls)

# aggregate using a circular window with radius 3
d <- mean(cont_ls, d = 3, type = "circle")
nomove_agg

Direct data aggregation

Description
Calculate the value for a given function for each cell in a larger resolution grid.

Usage
nomove_agg(coarse_dat, fine_dat, agg_fun, is_grid = TRUE, quiet = FALSE, ...)

Arguments
- **coarse_dat**: sf, Raster* or Spatial* object. The coarse grain data (response data) across which to calculate the aggregated function
- **fine_dat**: Raster* object. Raster* object. The fine grain data (predictor / covariate data) to aggregate
- **agg_fun**: function. The function to apply. The function fun should take multiple numbers, and return a single number. For example mean, modal, min or max. It should also accept a na.rm argument (or ignore it, e.g. as one of the 'dots' arguments. For example, length will fail, but function(x, ...)na.omit(length(x)) works. See Details
- **is_grid**: logical. Use TRUE (default) if g contains only rectangular cells (i.e. a grid). If g is any other polygon file, this should be set to false
- **quiet**: logical. If FALSE (default) and is_grid == TRUE the user gets a warning that the aggregation assumes all cells are rectangular
- **...**: further arguments passed to or from other methods

Details
grainchanger has several built-in functions. Functions currently included are:

- **shdi** - Shannon diversity, requires the additional argument lc_class (vector or scalar)
- **shei** - Shannon evenness, requires the additional argument lc_class (vector or scalar)
- **prop** - Proportion, requires the additional argument lc_class (scalar)
- **var_range** - Range (max - min)

Note that nomove_agg can be run in parallel using plan(multiprocess) from the future package.

Value
Raster (if input is Raster) or numeric vector (if input is sp or sf object) containing values calculated for each coarser cell
Examples

# load required data
data(g_sf)
data(cont_ls)
data(cat_ls)

# aggregate using mean
d <- nomove_agg(g_sf, cont_ls, mean)

# aggregate using Shannon evenness
d <- nomove_agg(g_sf, cont_ls, shei, lc_class = 1:4)

---

poly_sf

*Example polygon (coarse_dat)*

Description

An example non-gridded coarse data to show functionality when aggregating using an sf object.

Usage

poly_sf

Format

An sf object.

Details

Generated with `sf::st_make_grid(sf::st_as_sfc(sf::st_bbox(cont_ls)), cellsize = 13, square = FALSE)`

---

prop

*Calculate proportion of a given value*

Description

Calculate the proportion of a given value present within a raster. Useful for calculating land-cover or soil type proportions. Should be used with a categorical raster.

Usage

prop(x, lc_class, ...)

## S3 method for class 'winmove'
prop(x, lc_class, d, type, ...)

## S3 method for class 'numeric'
prop(x, lc_class, ...)
Arguments

x numeric, winmove. The data over which to calculate the proportion
lc_class numeric. The class value to calculate the proportion of
... further arguments passed to or from other methods
d numeric. If type=circle, the radius of the circle (in units of the CRS). If
type=rectangle the dimension of the rectangle (one or two numbers)
type character. The shape of the moving window

Value

If class(x) == "winmove", a smoothed raster with the proportion of cells of the given class calculated within the specified moving window
If class(x) == "numeric", a single value representing the proportion of values of a given class in x

Examples

# load required data
data(cat_ls)

# convert data to object of class winmove
cat_ls <- new("winmove", cat_ls)

# aggregate using a rectangular window with dimension 5 for class 3
d <- prop(cat_ls, d = 5, type = "rectangle", lc_class = 3)

# convert data to object of class numeric
cat_ls <- raster::values(cat_ls)
d <- prop(cat_ls, lc_class = 2)

---

var_range  Size of range of values

Description

Calculates the difference between the maximum and minimum value

Usage

var_range(x, ...)

## S3 method for class 'winmove'
var_range(x, d, type, na.rm = TRUE, ...)

## S3 method for class 'numeric'
var_range(x, na.rm = TRUE, ...)
Arguments

- \textbf{x} \hspace{1cm} \text{RasterLayer. The data over which to calculate the range size}
- ... \hspace{1cm} \text{further arguments passed to or from other methods}
- \textbf{d} \hspace{1cm} \text{numeric. If type=circle, the radius of the circle (in units of the CRS). If type=rectangle the dimension of the rectangle (one or two numbers)}
- \textbf{type} \hspace{1cm} \text{character. The shape of the moving window}
- \textbf{na.rm} \hspace{1cm} \text{logical. indicates whether NA values should be stripped before the computation proceeds. na.rm = TRUE is the default}

Value

If class(x) == "winmove", a smoothed raster with the size of the range of values calculated within the specified moving window.

If class(x) == "numeric", a single value representing the size of the range of values in \textit{x}.

Examples

```r
# load required data
data(cat_ls)
data(cont_ls)

# convert data to object of class winmove
cat_ls <- new("winmove", cat_ls)

# aggregate using a rectangular window with dimensions c(2,3)
d <- range(cont_ls, d = c(2,3), type = "rectangle")

# convert data to object of class numeric
cont_ls <- raster::values(cont_ls)
d <- range(cont_ls)
```

---

\textbf{winmove} \hspace{1cm} \textit{Create moving window surface}

Description

Smooth a raster surface using a moving window with a given function, radius and shape.

Usage

\texttt{winmove(fine_dat, d, type = c(\"circle\", \"rectangle\"), win_fun, ...)}
Arguments

- **fine_dat**: The raster dataset on which to calculate the moving window function
- **d**: numeric. If type=circle, the radius of the circle (in units of the CRS). If type=rectangle the dimension of the rectangle (one or two numbers).
- **type**: The shape of the moving window
- **win_fun**: function. The function to apply. If not choosing one of the inbuilt grainchanger functions, the function should take multiple numbers, and return a single number. For example mean, modal, min or max. It should also accept a na.rm argument (or ignore it, e.g. as one of the 'dots' arguments. For example, length will fail, but function(x,...)(na.omit(length(x))) works. See Details
  - ... further arguments passed to or from other methods

Details

grainchanger has several built-in functions. Functions currently included are:

- `wm_shei` - Shannon evenness, requires the additional argument `lc_class` (vector or scalar)
- `wm_prop` - Proportion, requires the additional argument `lc_class` (scalar)
- `wm_classes` - Unique number of classes in a categorical landscape
- `var_range` - Range (max - min)

Value

RasterLayer. A smoothed raster with the moving window values calculated

Examples

```r
# load required data
data(cat_ls)
data(cont_ls)

# calculate the moving window mean
d <- winmove(cont_ls, 5, "rectangle", mean)

# calculate the moving window Shannon evenness
d <- winmove(cat_ls, 5, "rectangle", shei, lc_class = 1:4)
```

---

**Description**

An S4 class for use with winmove functions (extends RasterLayer). Objects will need to be set to this class in order to be used with the inbuilt winmove functions (e.g. mean, prop, var_range, shdi, shei)
Slots

Slots for RasterLayer and RasterBrick objects

title: Character
file: Object of class ".RasterFile"
data: Object of class ".SingleLayerData" or ".MultipleLayerData"
history: To record processing history, not yet in use
legend: Object of class ".RasterLegend", Default legend. Should store preferences for plotting.
Not yet implemented except that it stores the color table of images, if available
extent: Object of 
ncols: Integer
nrows: Integer
crs: Object of class ".CRS", i.e. the coordinate reference system. In Spatial* objects this slot is called 'proj4string'

Examples

# load required data
data(cat_ls)

# set \code{cat_ls} to object of class winmove
new("winmove", cat_ls)

Description

Calculate the mean moving window value for a given radius, shape and function for each cell in a larger resolution grid.

Usage

```
winmove_agg(
  coarse_dat,
  fine_dat,
  d,
  type = c("circle", "rectangle"),
  win_fun,
  agg_fun = mean,
  is_grid = TRUE,
  quiet = FALSE,
  ...
)
```
Arguments

coarse_dat  sf, Raster* or Spatial* object. The coarse grain data (response data) across which to calculate the aggregated moving window function

fine_dat  Raster* object. The fine grain data (predictor / covariate data) to aggregate

d  numeric. If type=circle, the radius of the circle (in units of the CRS). If type=rectangle the dimension of the rectangle (one or two numbers).

type  character. The shape of the moving window

win_fun  character. The function to apply to the moving window. The function win_fun should take multiple numbers, and return a single number. For example: mean, modal, min or max. It should also accept a na.rm argument (or ignore it, e.g. as one of the 'dots' arguments. For example, length will fail, but function(x,...){na.omit(length(x))} works. See Details

agg_fun  character. The function by which to aggregate. By default this is set to mean

is_grid  logical. Use TRUE (default) if g contains only rectangular cells (i.e. a grid). If g is any other polygon file, this should be set to false

quiet  logical. If FALSE (default) and is_grid == TRUE the user gets a warning that the aggregation assumes all cells are rectangular

...  further arguments passed to or from other methods

Details

grainchanger has several built-in functions. Functions currently included are:

- shdi - Shannon diversity, requires the additional argument lc_class (vector or scalar)
- shei - Shannon evenness, requires the additional argument lc_class (vector or scalar)
- prop - Proportion, requires the additional argument lc_class (scalar)
- var_range - Range (max - min)

Note that winmove_agg can be run in parallel using plan(multiprocess) from the future package.

Value

Numeric vector containing moving window values calculated for each grid cell

Examples

```r
## Not run:
# load required data
data(g_sf)
data(cont_ls)
data(cat_ls)

# aggregate using mean
d <- winmove_agg(g_sf, cont_ls, 5, "rectangle", mean)

# aggregate using Shannon evenness
```
d <- winmove_agg(g.sf, cat.ls, 5, "rectangle", shei, lc_class = 1:4)

## End(Not run)
Index

* aggregate
  nomove_agg, 7
  winmove_agg, 12
* datasets
  cat_ls, 2
  cont_ls, 3
  g_sf, 5
  poly_sf, 8
* focal
  mean, 6
  prop, 8
  var_range, 9
  winmove, 10
  winmove_agg, 12
* mean
  mean, 6
  prop, 8
* range
  var_range, 9
* raster
  create_torus, 3
* spatial
  mean, 6
  nomove_agg, 7
  prop, 8
  var_range, 9
  winmove, 10
  winmove_agg, 12
* torus
  create_torus, 3

cat_ls, 2
cont_ls, 3
create_torus, 3

diversity-metrics, 4

g_sf, 5
mean, 6

nomove_agg, 7
poly_sf, 8
prop, 8
shdi(diversity-metrics), 4
shei(diversity-metrics), 4
var_range, 9
winmove, 10
winmove-class, 11
winmove_agg, 12