Package 'exactextractr'

October 12, 2021

Title Fast Extraction from Raster Datasets using Polygons

Version 0.7.1

Description Provides a replacement for the 'extract' function from the 'raster' package that is suitable for extracting raster values using 'sf' polygons.

Depends R (>= 3.4.0)

License Apache License (== 2.0)

SystemRequirements GEOS (>= 3.5.0)

Imports Rcpp (>= 0.12.12), methods, raster, sf (>= 0.9.0),

URL https://isciences.gitlab.io/exactextractr/, https://github.com/isciences/exactextractr

BugReports https://github.com/isciences/exactextractr/issues

LinkingTo Rcpp

Suggests dplyr, foreign, knitr, rgdal, rmarkdown, testthat, terra

Encoding UTF-8

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation yes

Author Daniel Baston [aut, cre], ISciences, LLC [cph]

Maintainer Daniel Baston <dbaston@isciences.com>

Repository CRAN

Date/Publication 2021-10-12 04:40:02 UTC

R topics documented:

coverage_fraction *Compute the fraction of raster cells covered by a polygon*

Description

Compute the fraction of raster cells covered by a polygon

Usage

```
## S4 method for signature 'RasterLayer,sf'
coverage_fraction(x, y, crop = FALSE)

## S4 method for signature 'RasterLayer,sfc_MULTIPOLYGON'
coverage_fraction(x, y, crop)

## S4 method for signature 'RasterLayer,sfc_POLYGON'
coverage_fraction(x, y, crop)
```

Arguments

x	a (possibly empty) RasterLayer whose resolution and extent will be used for the generated RasterLayer.
y	a sf object with polygonal geometries
crop	if TRUE, each generated RasterLayer will be cropped to the extent of its associated feature.

Value

a list with a RasterLayer for each feature in y. Values of the raster represent the fraction of each cell in x that is covered by y.

Examples

```
rast <- raster::raster(matrix(1:100, ncol=10), xmn=0, ymn=0, xmx=10, ymx=10)
poly <- sf::st_as_sfc('POLYGON ((2 2, 7 6, 4 9, 2 2))')

cov_frac <- coverage_fraction(rast, poly)[[1]]
```

exact_extract *Extract or summarize values from rasters*

Description

Extracts the values of cells in a raster (RasterLayer, RasterStack RasterBrick, or SpatRaster) that are covered by polygons in a simple feature collection (sf or sfc) or SpatialPolygonsDataFrame. Returns either a summary of the extracted values or the extracted values themselves.

Usage

```
## S4 method for signature 'Raster,sf'
exact_extract(
  x,
  y,
  fun = NULL,
  ...,
  weights = NULL,
  append_cols = NULL,
  coverage_area = FALSE,
  default_value = NA_real_,
  default_weight = NA_real_,
  include_area = FALSE,
  include_cell = FALSE,
  include_cols = NULL,
  include_xy = FALSE,
  force_df = FALSE,
  full_colnames = FALSE,
  stack_apply = FALSE,
  summarize_df = FALSE,
  quantiles = NULL,
  progress = TRUE,
  max_cells_in_memory = 3e+07
)

## S4 method for signature 'Raster,SpatialPolygonsDataFrame'
exact_extract(x, y, ...)

## S4 method for signature 'Raster,SpatialPolygons'
exact_extract(x, y, ...)

## S4 method for signature 'Raster,sfc_MULTIPOLYGON'
exact_extract(
  x,
  y,
  fun = NULL,
  ...,
```

```
  weights = NULL,
  append_cols = NULL,
  coverage_area = FALSE,
  default_value = NA_real_,
  default_weight = NA_real_,
  include_area = FALSE,
  include_cell = FALSE,
  include_cols = NULL,
  include_xy = FALSE,
  force_df = FALSE,
  full_colnames = FALSE,
  stack_apply = FALSE,
  summarize_df = FALSE,
  quantiles = NULL,
  progress = TRUE,
  max_cells_in_memory = 3e+07
)

## S4 method for signature 'Raster,sfc_POLYGON'
exact_extract(
  x,
  y,
  fun = NULL,
  ...,
  weights = NULL,
  append_cols = NULL,
  coverage_area = FALSE,
  default_value = NA_real_,
  default_weight = NA_real_,
  include_area = FALSE,
  include_cell = FALSE,
  include_cols = NULL,
  include_xy = FALSE,
  force_df = FALSE,
  full_colnames = FALSE,
  stack_apply = FALSE,
  summarize_df = FALSE,
  quantiles = NULL,
  progress = TRUE,
  max_cells_in_memory = 3e+07
)

## S4 method for signature 'Raster,sfc_GEOMETRY'
exact_extract(
  x,
  y,
  fun = NULL,
  ...,
```

```
  weights = NULL,
  append_cols = NULL,
  coverage_area = FALSE,
  default_value = NA_real_,
  default_weight = NA_real_,
  include_area = FALSE,
  include_cell = FALSE,
  include_cols = NULL,
  include_xy = FALSE,
  force_df = FALSE,
  full_colnames = FALSE,
  stack_apply = FALSE,
  summarize_df = FALSE,
  quantiles = NULL,
  progress = TRUE,
  max_cells_in_memory = 3e+07
)

## S4 method for signature 'Raster,sfc_GEOMETRYCOLLECTION'
exact_extract(
  x,
  y,
  fun = NULL,
  ...,
  weights = NULL,
  append_cols = NULL,
  coverage_area = FALSE,
  default_value = NA_real_,
  default_weight = NA_real_,
  include_area = FALSE,
  include_cell = FALSE,
  include_cols = NULL,
  include_xy = FALSE,
  force_df = FALSE,
  full_colnames = FALSE,
  stack_apply = FALSE,
  summarize_df = FALSE,
  quantiles = NULL,
  progress = TRUE,
  max_cells_in_memory = 3e+07
)

## S4 method for signature 'SpatRaster,sf'
exact_extract(
  x,
  y,
  fun = NULL,
  ...,
```

```
  weights = NULL,
  append_cols = NULL,
  coverage_area = FALSE,
  default_value = NA_real_,
  default_weight = NA_real_,
  include_area = FALSE,
  include_cell = FALSE,
  include_cols = NULL,
  include_xy = FALSE,
  force_df = FALSE,
  full_colnames = FALSE,
  stack_apply = FALSE,
  summarize_df = FALSE,
  quantiles = NULL,
  progress = TRUE,
  max_cells_in_memory = 3e+07
)

## S4 method for signature 'SpatRaster,SpatialPolygonsDataFrame'
exact_extract(x, y, ...)

## S4 method for signature 'SpatRaster,SpatialPolygons'
exact_extract(x, y, ...)

## S4 method for signature 'SpatRaster,sfc_MULTIPOLYGON'
exact_extract(
  x,
  y,
  fun = NULL,
  ...,
  weights = NULL,
  append_cols = NULL,
  coverage_area = FALSE,
  default_value = NA_real_,
  default_weight = NA_real_,
  include_area = FALSE,
  include_cell = FALSE,
  include_cols = NULL,
  include_xy = FALSE,
  force_df = FALSE,
  full_colnames = FALSE,
  stack_apply = FALSE,
  summarize_df = FALSE,
  quantiles = NULL,
  progress = TRUE,
  max_cells_in_memory = 3e+07
)
```

```
## S4 method for signature 'SpatRaster,sfc_POLYGON'
exact_extract(
  x,
  y,
  fun = NULL,
  ...,
  weights = NULL,
  append_cols = NULL,
  coverage_area = FALSE,
  default_value = NA_real_,
  default_weight = NA_real_,
  include_area = FALSE,
  include_cell = FALSE,
  include_cols = NULL,
  include_xy = FALSE,
  force_df = FALSE,
  full_colnames = FALSE,
  stack_apply = FALSE,
  summarize_df = FALSE,
  quantiles = NULL,
  progress = TRUE,
  max_cells_in_memory = 3e+07
)

## S4 method for signature 'SpatRaster,sfc_GEOMETRY'
exact_extract(
  x,
  y,
  fun = NULL,
  ...,
  weights = NULL,
  append_cols = NULL,
  coverage_area = FALSE,
  default_value = NA_real_,
  default_weight = NA_real_,
  include_area = FALSE,
  include_cell = FALSE,
  include_cols = NULL,
  include_xy = FALSE,
  force_df = FALSE,
  full_colnames = FALSE,
  stack_apply = FALSE,
  summarize_df = FALSE,
  quantiles = NULL,
  progress = TRUE,
  max_cells_in_memory = 3e+07
)
```

```
## S4 method for signature 'SpatRaster,sfc_GEOMETRYCOLLECTION'
exact_extract(
  x,
  y,
  fun = NULL,
  ...,
  weights = NULL,
  append_cols = NULL,
  coverage_area = FALSE,
  default_value = NA_real_,
  default_weight = NA_real_,
  include_area = FALSE,
  include_cell = FALSE,
  include_cols = NULL,
  include_xy = FALSE,
  force_df = FALSE,
  full_colnames = FALSE,
  stack_apply = FALSE,
  summarize_df = FALSE,
  quantiles = NULL,
  progress = TRUE,
  max_cells_in_memory = 3e+07
)
```

Arguments

x	a RasterLayer, RasterStack, RasterBrick, or SpatRaster
y	a sf, sfc, SpatialPolygonsDataFrame, or SpatialPolygons object with polygonal geometries
fun	an optional function or character vector, as described below
...	additional arguments to pass to fun
weights	a weighting raster to be used with the weighted_mean and weighted_sum summary operations or a user-defined summary function. When weights is set to 'area', the cell areas of x will be calculated and used as weights.
append_cols	when fun is not NULL, an optional character vector of columns from y to be included in returned data frame.
coverage_area	if TRUE, output pixel coverage_area instead of coverage_fraction
default_value	an optional value to use instead of NA in x
default_weight	an optional value to use instead of NA in weights
include_area	if TRUE, and fun is NULL, augment the data frame for each feature with a column for the cell area. If the units of the raster CRS are degrees, the area in square meters will be calculated based on a spherical approximation of Earth. Otherwise, a Cartesian area will be calculated (and will be the same for all pixels.) If TRUE and fun is not NULL, add area to the data frame passed to fun for each feature.

include_cell — if TRUE, and fun is NULL, augment the data frame for each feature with a column for the cell index (cell). If TRUE and fun is not NULL, add cell to the data frame passed to fun for each feature.

include_cols — an optional character vector of column names in y to be added to the data frame for each feature that is either returned (when fun is NULL) or passed to fun.

include_xy — if TRUE, and fun is NULL, augment the returned data frame for each feature with columns for cell center coordinates (x and y). If TRUE and fun is not NULL, add x and y to the data frame passed to fun for each feature.

force_df — always return a data frame instead of a vector, even if x has only one layer and fun has length 1

full_colnames — include the names of x and weights in the names of the data frame for each feature, even if x or weights has only one layer. This is useful when the results of multiple calls to exact_extract are combined with cbind.

stack_apply — if TRUE, apply fun independently to each layer or x (and its corresponding layer of weights, if provided.) The number of layers in x and weights must equal each other or 1, in which case the single layer raster will be recycled. If FALSE, apply fun to all layers of x (and weights) simultaneously.

summarize_df — pass values, coverage fraction/area, and weights to fun as a single data frame instead of separate arguments.

quantiles — quantiles to be computed when fun = 'quantile'

progress — if TRUE, display a progress bar during processing

max_cells_in_memory — the maximum number of raster cells to load at a given time when using a named summary operation for fun (as opposed to a function defined using R code). If a polygon covers more than max_cells_in_memory raster cells, it will be processed in multiple chunks.

Details

exact_extract extracts the values of cells in a raster that are covered by polygonal features in a simple feature collection (sf or sfc) or SpatialPolygonDataFrame, as well as the fraction or area of each cell that is covered by the feature. Pixels covered by all parts of the polygon are considered. If an (invalid) multipart polygon covers the same pixels more than once, the pixel may have a coverage fraction greater than one.

The function can either return pixel values directly to the caller, or can return the result of a predefined summary operation or user-defined R function applied to the values. These three approaches are described in the subsections below.

Returning extracted values directly:

If fun is not specified, exact_extract will return a list with one data frame for each feature in the input feature collection. The data frame will contain a column with cell values from each layer in the input raster (and optional weighting raster) and a column indicating the fraction or area of the cell that is covered by the polygon.

If the input rasters have only one layer, the value and weight columns in the data frame will be named values or weights. When the input rasters have more than one layer, the columns will

be named according to `names(x)` and `names(weights)`. The column containing pixel coverage will be called `coverage_fraction` when `coverage_area = FALSE`, or `coverage_area` when `coverage_area = TRUE`. Additional columns can be added to the returned data frames with the `include_area`, `include_cell`, and `include_xy` arguments.

If the output data frames for multiple features are to be combined (e.g., with `rbind`), it may be useful to include identifying column(s) from the input features in the returned data frames using `include_cols`.

Predefined summary operations:

Often the individual pixel values are not needed; only one or more summary statistics (e.g., mean, sum) is required for each feature. Common summary statistics can be calculated by `exact_extract` directly using a predefined summary operation. Where possible, this approach is advantageous because it allows the package to calculate the statistics incrementally, avoiding the need to store all pixel values in memory at the same time. This allows the package to process arbitrarily large data with a small amount of memory. (The `max_pixels_in_memory` argument can be used to fine-tune the amount of memory made available to `exact_extract`.)

To summarize pixel values using a predefined summary option, `fun` should be set to a character vector of one or more operation names. If the input raster has a single layer and a single summary operation is specified, `exact_extract` will return a vector with the result of the summary operation for each feature in the input. If the input raster has multiple layers, or if multiple summary operations are specified, `exact_extract` will return a data frame with a row for each feature and a column for each summary operation / layer combination. (The `force_df` option can be used to always return a data frame instead of a vector.)

The following summary operations are supported:

- `min` - the minimum non-`NA` value in any raster cell wholly or partially covered by the polygon
- `max` - the maximum non-`NA` value in any raster cell wholly or partially covered by the polygon
- `count` - the sum of fractions of raster cells with non-`NA` values covered by the polygon
- `sum` - the sum of non-`NA` raster cell values, multiplied by the fraction of the cell that is covered by the polygon
- `mean` - the mean cell value, weighted by the fraction of each cell that is covered by the polygon
- `median` - the median cell value, weighted by the fraction of each cell that is covered by the polygon
- `quantile` - arbitrary quantile(s) of cell values, specified in `quantiles`, weighted by the fraction of each cell that is covered by the polygon
- `mode` - the most common cell value, weighted by the fraction of each cell that is covered by the polygon. Where multiple values occupy the same maximum number of weighted cells, the largest value will be returned.
- `majority` - synonym for `mode`
- `minority` - the least common cell value, weighted by the fraction of each cell that is covered by the polygon. Where multiple values occupy the same minimum number of weighted cells, the smallest value will be returned.
- `variety` - the number of distinct values in cells that are wholly or partially covered by the polygon.
- `variance` - the population variance of cell values, weighted by the fraction of each cell that is covered by the polygon.

- stdev - the population standard deviation of cell values, weighted by the fraction of each cell that is covered by the polygon.
- coefficient_of_variation - the population coefficient of variation of cell values, weighted by the fraction of each cell that is covered by the polygon.
- weighted_mean - the mean cell value, weighted by the product of the fraction of each cell covered by the polygon and the value of a second weighting raster provided as weights
- weighted_sum - the sum of defined raster cell values, multiplied by the fraction of each cell that is covered by the polygon and the value of a second weighting raster provided as weights

In all of the summary operations, NA values in the the primary raster (x) raster are ignored (i.e., na.rm = TRUE.) If NA values occur in the weighting raster, the result of the weighted operation will be NA. NA values in both x and weights can be replaced on-the-fly using the default_value and default_weight arguments.

User-defined summary functions:

If no predefined summary operation is suitable, a user-defined R function may be provided as fun. The function will be called once for each feature and must return either a single value or a data frame. The results of the function for each feature will be combined and returned by exact_extract.

The simplest way to write a summary function is to set argument summarize_df = TRUE. (For backwards compatibility, this is not the default.) In this mode, the summary function takes the signature function(df, ...) where df is the same data frame that would be returned by exact_extract with fun = NULL.

With summarize_df = FALSE, the function must have the signature function(values, coverage_fractions, ...) when weights are not used, and function(values, coverage_fractions, weights, ...) when weights are used. If the value and weight rasters each have a single layer, the function arguments will be vectors; if either has multiple layers, the function arguments will be data frames, with column names taken from the names of the value/weight rasters. Values brought in through the include_xy, include_area, include_cell, and include_cols arguments will be added to the values data frame. For most applications, it is simpler to set summarize_df = TRUE and work with all inputs in a single data frame.

Value

a vector, data frame, or list of data frames, depending on the type of x and the value of fun (see Details)

Examples

```
rast <- raster::raster(matrix(1:100, ncol=10), xmn=0, ymn=0, xmx=10, ymx=10)
poly <- sf::st_as_sfc('POLYGON ((2 2, 7 6, 4 9, 2 2))')

# named summary operation on RasterLayer, returns vector
exact_extract(rast, poly, 'mean')

# two named summary operations on RasterLayer, returns data frame
exact_extract(rast, poly, c('min', 'max'))

# named summary operation on RasterStack, returns data frame
stk <- raster::stack(list(a=rast, b=sqrt(rast)))
```

```
exact_extract(stk, poly, 'mean')

# named weighted summary operation, returns vector
weights <- raster::raster(matrix(runif(100), ncol=10), xmn=0, ymn=0, xmx=10, ymx=10)
exact_extract(rast, poly, 'weighted_mean', weights=weights)

# custom summary function, returns vector
exact_extract(rast, poly, function(value, cov_frac) length(value[cov_frac > 0.9]))
```

`exact_resample` *Resample a raster to a new grid*

Description

Resample a raster to a new grid

Usage

```
## S4 method for signature 'RasterLayer,RasterLayer'
exact_resample(x, y, fun)

## S4 method for signature 'SpatRaster,SpatRaster'
exact_resample(x, y, fun)
```

Arguments

`x`	a `RasterLayer` or `SpatRaster` to be resampled
`y`	a raster of the same class as `x` with a grid definition to which `x` should be resampled
`fun`	a named summary operation to be used for the resampling

Value

a resampled version of `x`, returned as a `RasterLayer` or `SpatRaster`, depending on the values of `x` and `y`

Index