# Package ‘landscapemetrics’

July 8, 2020

**Type** Package

**Title** Landscape Metrics for Categorical Map Patterns

**Version** 1.4.5

**Maintainer** Maximillian H.K. Hesselbarth <mhk.hesselbarth@gmail.com>

**Description** Calculates landscape metrics for categorical landscape patterns in
a tidy workflow. 'landscapemetrics' reimplements the most common metrics from
'FRAGSTATS' (<https://www.umass.edu/landeco/research/fragstats/fragstats.html>)
and new ones from the current literature on landscape metrics.
This package supports 'raster' spatial objects and takes
RasterLayer, RasterStacks, RasterBricks or lists of RasterLayer from the
'raster' package as input arguments. It further provides utility functions
to visualize patches, select metrics and building blocks to develop new
metrics.

**License** GPL-3

**URL** https://r-spatialecology.github.io/landscapemetrics/

**BugReports** https://github.com/r-spatialecology/landscapemetrics/issues

**Depends** R (>= 3.1)

**Imports** cli, ggplot2, methods, raster, Rcpp, sp, stats, tibble

**Suggests** covr, dplyr, knitr, rgeos, rmarkdown, testthat

**Enhances** stars, sf

**LinkingTo** Rcpp, RcppArmadillo

**ByteCompile** true

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1.9000

**SystemRequirements** C++11

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Maximillian H.K. Hesselbarth [aut, cre]
(<https://orcid.org/0000-0003-1125-9918>),
Marco Sciaini [aut] (<https://orcid.org/0000-0002-3042-5435>),
Jakub Nowosad [aut] (<https://orcid.org/0000-0002-1057-3721>),
Sebastian Hanss [aut] (<https://orcid.org/0000-0002-3990-4897>),
Laura J. Graham [ctb] (Input on package structure),
Jeffrey Hollister [ctb] (Input on package structure),
Kimberly A. With [ctb] (Input on package structure),
Florian Privé [ctb] (Original author of underlying C++ code for
get_nearestneighbour() function),
Project Nayuki [ctb] (Original author of underlying C++ code for
get_circumscribingcircle and lsm_p_circle),
Matt Strimas-Mackey [ctb] (Bugfix in sample_metrics())

**Repository** CRAN

**Date/Publication** 2020-07-08 13:30:02 UTC

# R topics documented:

augusta_nlcd . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
calculate_correlation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
calculate_lsm . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
check_landscape . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
extract_lsm . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
fragstats_class_augusta_nlcd . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
fragstats_class_landscape . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
fragstats_class_podlasie . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
fragstats_landscape_augusta_nlcd . . . . . . . . . . . . . . . . . . . . . . . . . 17
fragstats_landscape_landscape . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
fragstats_landscape_podlasie . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
fragstats_patch_augusta_nlcd . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
fragstats_patch_landscape . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
fragstats_patch_podlasie . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
get_adjacencies . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
get_boundaries . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
get_centroids . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
get_circumscribingcircle . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
get_nearestneighbour . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
get_patches . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
get_unique_values . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
landscape . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
landscapemetrics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
list_lsm . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
lsm_abbreviations_names . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
lsm_c_ai . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
lsm_c_area_cv . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
lsm_c_area_mn . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
lsm_c_area_sd . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

lsm_c_ca . . . . . 41
lsm_c_cai_cv . . . . . 42
lsm_c_cai_mn . . . . . 44
lsm_c_cai_sd . . . . . 46
lsm_c_circle_cv . . . . . 48
lsm_c_circle_mn . . . . . 50
lsm_c_circle_sd . . . . . 52
lsm_c_clumpy . . . . . 53
lsm_c_cohesion . . . . . 55
lsm_c_contig_cv . . . . . 56
lsm_c_contig_mn . . . . . 58
lsm_c_contig_sd . . . . . 59
lsm_c_core_cv . . . . . 61
lsm_c_core_mn . . . . . 63
lsm_c_core_sd . . . . . 65
lsm_c_cpland . . . . . 67
lsm_c_dcad . . . . . 69
lsm_c_dcore_cv . . . . . 72
lsm_c_dcore_mn . . . . . 74
lsm_c_dcore_sd . . . . . 76
lsm_c_division . . . . . 78
lsm_c_ed . . . . . 80
lsm_c_enn_cv . . . . . 81
lsm_c_enn_mn . . . . . 83
lsm_c_enn_sd . . . . . 84
lsm_c_frac_cv . . . . . 86
lsm_c_frac_mn . . . . . 88
lsm_c_frac_sd . . . . . 89
lsm_c_gyrate_cv . . . . . 91
lsm_c_gyrate_mn . . . . . 92
lsm_c_gyrate_sd . . . . . 94
lsm_c_iji . . . . . 96
lsm_c_lpi . . . . . 97
lsm_c_lsi . . . . . 99
lsm_c_mesh . . . . . 100
lsm_c_ndca . . . . . 102
lsm_c_nlsi . . . . . 104
lsm_c_np . . . . . 105
lsm_c_pafrac . . . . . 107
lsm_c_para_cv . . . . . 108
lsm_c_para_mn . . . . . 110
lsm_c_para_sd . . . . . 111
lsm_c_pd . . . . . 113
lsm_c_pladj . . . . . 114
lsm_c_pland . . . . . 115
lsm_c_shape_cv . . . . . 117
lsm_c_shape_mn . . . . . 119
lsm_c_shape_sd . . . . . 120

lsm_c_split . . . . . 122
lsm_c_tca . . . . . 123
lsm_c_te . . . . . 125
lsm_l_ai . . . . . 126
lsm_l_area_cv . . . . . 128
lsm_l_area_mn . . . . . 129
lsm_l_area_sd . . . . . 131
lsm_l_cai_cv . . . . . 132
lsm_l_cai_mn . . . . . 134
lsm_l_cai_sd . . . . . 136
lsm_l_circle_cv . . . . . 138
lsm_l_circle_mn . . . . . 140
lsm_l_circle_sd . . . . . 142
lsm_l_cohesion . . . . . 143
lsm_l_condent . . . . . 145
lsm_l_contag . . . . . 146
lsm_l_contig_cv . . . . . 148
lsm_l_contig_mn . . . . . 149
lsm_l_contig_sd . . . . . 151
lsm_l_core_cv . . . . . 153
lsm_l_core_mn . . . . . 155
lsm_l_core_sd . . . . . 157
lsm_l_dcad . . . . . 159
lsm_l_dcore_cv . . . . . 161
lsm_l_dcore_mn . . . . . 163
lsm_l_dcore_sd . . . . . 165
lsm_l_division . . . . . 167
lsm_l_ed . . . . . 169
lsm_l_enn_cv . . . . . 170
lsm_l_enn_mn . . . . . 172
lsm_l_enn_sd . . . . . 173
lsm_l_ent . . . . . 175
lsm_l_frac_cv . . . . . 176
lsm_l_frac_mn . . . . . 178
lsm_l_frac_sd . . . . . 179
lsm_l_gyrate_cv . . . . . 181
lsm_l_gyrate_mn . . . . . 182
lsm_l_gyrate_sd . . . . . 184
lsm_l_iji . . . . . 186
lsm_l_joinent . . . . . 187
lsm_l_lpi . . . . . 188
lsm_l_lsi . . . . . 190
lsm_l_mesh . . . . . 191
lsm_l_msidi . . . . . 193
lsm_l_msiei . . . . . 195
lsm_l_mutinf . . . . . 196
lsm_l_ndca . . . . . 197
lsm_l_np . . . . . 200

lsm_l_pafrac . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 201
lsm_l_para_cv . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 203
lsm_l_para_mn . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 204
lsm_l_para_sd . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 206
lsm_l_pd . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 207
lsm_l_pladj . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 209
lsm_l_pr . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 210
lsm_l_prd . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 211
lsm_l_rpr . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 213
lsm_l_shape_cv . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 214
lsm_l_shape_mn . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 216
lsm_l_shape_sd . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 217
lsm_l_shdi . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 219
lsm_l_shei . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 220
lsm_l_sidi . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 222
lsm_l_siei . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 223
lsm_l_split . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 225
lsm_l_ta . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 226
lsm_l_tca . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 228
lsm_l_te . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 229
lsm_p_area . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 231
lsm_p_cai . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 232
lsm_p_circle . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 234
lsm_p_contig . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 235
lsm_p_core . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 237
lsm_p_enn . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 239
lsm_p_frac . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 240
lsm_p_gyrate . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 242
lsm_p_ncore . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 243
lsm_p_para . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 245
lsm_p_perim . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 247
lsm_p_shape . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 248
options_landscapemetrics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 250
podlasie_ccilc . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 250
sample_lsm . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 251
scale_sample . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 254
scale_window . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 257
show_cores . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 259
show_correlation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 262
show_lsm . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 263
show_patches . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 266
spatialize_lsm . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 267
window_lsm . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 270

**Index** **274**

## augusta_nlcd *Augusta NLCD 2011*

### Description

A real landscape of area near Augusta, Georgia obtained from the National Land Cover Database (NLCD)

### Usage

```
augusta_nlcd
```

### Format

A raster layer object.

### Source

https://www.mrlc.gov/nlcd2011.php

### References

Homer, C.G., Dewitz, J.A., Yang, L., Jin, S., Danielson, P., Xian, G., Coulston, J., Herold, N.D., Wickham, J.D., and Megown, K., 2015, Completion of the 2011 National Land Cover Database for the conterminous United States-Representing a decade of land cover change information. Photogrammetric Engineering and Remote Sensing, v. 81, no. 5, p. 345-354

## calculate_correlation *Calculate correlation*

### Description

Calculate correlation

### Usage

```
calculate_correlation(
  metrics,
  method = "pearson",
  diag = TRUE,
  simplify = FALSE
)
```

### Arguments

| | |
|---|---|
| `metrics` | Tibble with results of as returned by the landscapemetrics package. |
| `method` | Type of correlation. See `link{cor}` for details. |
| `diag` | If FALSE, values on the diagonal will be NA. |
| `simplify` | If TRUE and only one level is present, only a tibble is returned. |

### Details

The functions calculates the correlation between all metrics. In order to calculate correlations, for the landscape level more than one landscape needs to be present. All input must be structured as returned by the **landscapemetrics** package.

### Value

list

### Examples

```
metrics <- calculate_lsm(landscape, what = c("patch", "class"))
calculate_correlation(metrics, method = "pearson")
```

## `calculate_lsm` *calculate_lsm*

### Description

Calculate a selected group of metrics

### Usage

```
calculate_lsm(
  landscape,
  level,
  metric,
  name,
  type,
  what,
  directions,
  count_boundary,
  consider_boundary,
  edge_depth,
  cell_center,
  classes_max,
  neighbourhood,
  ordered,
  base,
  full_name,
  verbose,
  progress
)

## S3 method for class 'RasterLayer'
calculate_lsm(
  landscape,
  level = NULL,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  count_boundary = FALSE,
  consider_boundary = FALSE,
  edge_depth = 1,
  cell_center = FALSE,
  classes_max = NULL,
  neighbourhood = 4,
  ordered = TRUE,
  base = "log2",
  full_name = FALSE,
  verbose = TRUE,
  progress = FALSE
)

## S3 method for class 'RasterStack'
calculate_lsm(
  landscape,
  level = NULL,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  count_boundary = FALSE,
  consider_boundary = FALSE,
  edge_depth = 1,
  cell_center = FALSE,
  classes_max = NULL,
  neighbourhood = 4,
  ordered = TRUE,
  base = "log2",
  full_name = FALSE,
  verbose = TRUE,
  progress = FALSE
```

```
)

## S3 method for class 'RasterBrick'
calculate_lsm(
  landscape,
  level = NULL,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  count_boundary = FALSE,
  consider_boundary = FALSE,
  edge_depth = 1,
  cell_center = FALSE,
  classes_max = NULL,
  neighbourhood = 4,
  ordered = TRUE,
  base = "log2",
  full_name = FALSE,
  verbose = TRUE,
  progress = FALSE
)

## S3 method for class 'stars'
calculate_lsm(
  landscape,
  level = NULL,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  count_boundary = FALSE,
  consider_boundary = FALSE,
  edge_depth = 1,
  cell_center = FALSE,
  classes_max = NULL,
  neighbourhood = 4,
  ordered = TRUE,
  base = "log2",
  full_name = FALSE,
  verbose = TRUE,
  progress = FALSE
)

## S3 method for class 'list'
calculate_lsm(
  landscape,
  level = NULL,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  count_boundary = FALSE,
  consider_boundary = FALSE,
  edge_depth = 1,
  cell_center = FALSE,
  classes_max = NULL,
  neighbourhood = 4,
  ordered = TRUE,
  base = "log2",
  full_name = FALSE,
  verbose = TRUE,
  progress = FALSE
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`level` Level of metrics. Either 'patch', 'class' or 'landscape' (or vector with combination).

`metric` Abbreviation of metrics (e.g. 'area').

`name` Full name of metrics (e.g. 'core area').

`type` Type according to FRAGSTATS grouping (e.g. 'aggregation metrics').

`what` Selected level of metrics: either "patch", "class" or "landscape". It is also possible to specify functions as a vector of strings, e.g. `what = c("lsm_c_ca","lsm_l_ta")`.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`count_boundary` Include landscape boundary in edge length.

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core.

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell.

`cell_center` If true, the coordinates of the centroid are forced to be a cell center within the patch.

`classes_max` Potential maximum number of present classes.

`neighbourhood` The number of directions in which cell adjacencies are considered as neighbours: 4 (rook's case) or 8 (queen's case). The default is 4.

`ordered` The type of pairs considered. Either ordered (TRUE) or unordered (FALSE). The default is TRUE.

`base` The unit in which entropy is measured. The default is "log2", which compute entropy in "bits". "log" and "log10" can be also used.

`full_name` Should the full names of all functions be included in the tibble.

`verbose` Print warning messages.

`progress` Print progress report.

## Details

Wrapper to calculate several landscape metrics. The metrics can be specified by the arguments `what`, `level`, `metric`, `name` and/or `type` (combinations of different arguments are possible (e.g. level = "class", type = "aggregation metric"). If an argument is not provided, automatically all possibilities are selected. Therefore, to get **all** available metrics, don't specify any of the above arguments.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`list_lsm`

## Examples

```
## Not run:
calculate_lsm(landscape, progress = TRUE)
calculate_lsm(landscape, what = c("patch", "lsm_c_te", "lsm_l_pr"))
calculate_lsm(landscape, level = c("class", "landscape"), type = "aggregation metric")

## End(Not run)
```

# `check_landscape` *Check input landscape*

## Description

Check input landscape

### Usage

```
check_landscape(landscape, verbose)

## S3 method for class 'RasterLayer'
check_landscape(landscape, verbose = TRUE)

## S3 method for class 'RasterStack'
check_landscape(landscape, verbose = TRUE)

## S3 method for class 'RasterBrick'
check_landscape(landscape, verbose = TRUE)

## S3 method for class 'stars'
check_landscape(landscape, verbose = TRUE)

## S3 method for class 'list'
check_landscape(landscape, verbose = TRUE)
```

### Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick, Stars or a list of rasterLayers |
| verbose | Print warning messages. |

### Details

This function extracts basic information about the input landscape. It includes a type of coordinate reference system (crs) - either "geographic", "projected", or NA, units of the coordinate reference system, a class of the input landscape's values and the number of classes found in the landscape.

### Value

tibble

### Examples

```
check_landscape(augusta_nlcd)
check_landscape(podlasie_ccilc)
check_landscape(raster::stack(landscape, landscape))
```

---

## extract_lsm *extract_lsm*

---

### Description

Extract metrics

### Usage

```
extract_lsm(
  landscape,
  y,
  extract_id,
  metric,
  name,
  type,
  what,
  directions,
  progress,
  verbose,
  ...
)

## S3 method for class 'RasterLayer'
extract_lsm(
  landscape,
  y,
  extract_id = NULL,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  progress = FALSE,
  verbose = TRUE,
  ...
)

## S3 method for class 'RasterStack'
extract_lsm(
  landscape,
  y,
  extract_id = NULL,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  progress = FALSE,
  verbose = TRUE,
  ...
)

## S3 method for class 'RasterBrick'
extract_lsm(
  landscape,
  y,
  extract_id = NULL,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  progress = FALSE,
  verbose = TRUE,
  ...
)

## S3 method for class 'stars'
extract_lsm(
  landscape,
  y,
  extract_id = NULL,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  progress = FALSE,
  verbose = TRUE,
  ...
)

## S3 method for class 'list'
extract_lsm(
  landscape,
  y,
  extract_id = NULL,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  progress = FALSE,
  verbose = TRUE,
  ...
)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `y` | 2-column matrix with coordinates, SpatialPoints, SpatialLines or sf point geometries. |
| `extract_id` | Vector with id of sample points. If not provided, sample points will be labelled |

1...n.

`metric` Abbreviation of metrics (e.g. 'area').

`name` Full name of metrics (e.g. 'core area')

`type` Type according to FRAGSTATS grouping (e.g. 'aggregation metrics').

`what` Selected level of metrics: either "patch", "class" or "landscape". It is also possible to specify functions as a vector of strings, e.g. `what = c("lsm_c_ca","lsm_l_ta")`.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`progress` Print progress report.

`verbose` Print warning messages.

`...` Arguments passed to `calculate_lsm()`.

## Details

This functions extracts the metrics of all patches the spatial object(s) `y` (e.g. spatial points) are located within. Only patch level metrics are possible to extract. Please be aware that the output is sligthly different to all other `lsm`-function of `landscapemetrics`. Returns a tibble with chosen metrics and the ID of the spatial objects.

## Value

tibble

## See Also

`calculate_lsm`

## Examples

```
points <- matrix(c(10, 5, 25, 15, 5, 25), ncol = 2, byrow = TRUE)
extract_lsm(landscape, y = points)
extract_lsm(landscape, y = points, type = "aggregation metric")

points_sp <- sp::SpatialPoints(points)
extract_lsm(landscape, y = points_sp, what = "lsm_p_area")

## Not run:
# use lines (works only if rgeos is installed)
x1 <- c(1, 5, 15, 10)
y1 <- c(1, 5, 15, 25)

x2 <- c(10, 25)
y2 <- c(5, 5)

sample_lines <- sp::SpatialLines(list(sp::Lines(list(sp::Line(cbind(x1, y1)),
sp::Line(cbind(x2, y2))), ID = "a")))
extract_lsm(landscape, y = sample_lines, what = "lsm_p_area")

## End(Not run)
```

`fragstats_class_augusta_nlcd`

*Fragstats results for landscapemetrics::augusta_nlcd (class level)*

## Description

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on class level.

## Usage

```
fragstats_class_augusta_nlcd
```

## Format

A tibble object.

`fragstats_class_landscape`

*Fragstats results for landscapemetrics::landscape (class level)*

## Description

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on class level.

## Usage

```
fragstats_class_landscape
```

## Format

A tibble object.

`fragstats_class_podlasie` *Fragstats results for landscapemetrics::podlasie (class level)*

### Description

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on class level.

### Usage

```
fragstats_class_podlasie
```

### Format

A tibble object.

`fragstats_landscape_augusta_nlcd` *Fragstats results for landscapemetrics::augusta_nlcd (landscape level)*

### Description

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on landscape level.

### Usage

```
fragstats_landscape_augusta_nlcd
```

### Format

A tibble object.

`fragstats_landscape_landscape`

*Fragstats results for landscapemetrics::landscape (landscape level)*

## Description

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on landscape level.

## Usage

```
fragstats_landscape_landscape
```

## Format

A tibble object.

`fragstats_landscape_podlasie`

*Fragstats results for landscapemetrics::podlasie_ccilc (landscape level)*

## Description

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on landscape level.

## Usage

```
fragstats_landscape_podlasie
```

## Format

A tibble object.

## fragstats_patch_augusta_nlcd *Fragstats results for landscapemetrics::augusta_nlcd (patch level)*

### Description

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on patch level.

### Usage

```
fragstats_patch_augusta_nlcd
```

### Format

A tibble object.

## fragstats_patch_landscape *Fragstats results for landscapemetrics::landscape (patch level)*

### Description

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on patch level.

### Usage

```
fragstats_patch_landscape
```

### Format

A tibble object.

## fragstats_patch_podlasie *Fragstats results for landscapemetrics::podlasie (patch level)*

### Description

A single tibble for every spatial dataset included in landscapemetrics that contains the FRAGSTAT results for every implemented metric on patch level.

### Usage

```
fragstats_patch_podlasie
```

### Format

A tibble object.

## get_adjacencies *get_adjacencies*

### Description

Fast calculation of adjacencies between classes in a raster

### Usage

```
get_adjacencies(landscape, neighbourhood, what, upper)

## S3 method for class 'RasterLayer'
get_adjacencies(landscape, neighbourhood = 4, what = "full", upper = FALSE)

## S3 method for class 'RasterStack'
get_adjacencies(landscape, neighbourhood = 4, what = "full", upper = FALSE)

## S3 method for class 'RasterBrick'
get_adjacencies(landscape, neighbourhood = 4, what = "full", upper = FALSE)

## S3 method for class 'stars'
get_adjacencies(landscape, neighbourhood = 4, what = "full", upper = FALSE)

## S3 method for class 'list'
get_adjacencies(landscape, neighbourhood = 4, what = "full", upper = FALSE)

## S3 method for class 'matrix'
get_adjacencies(landscape, neighbourhood = 4, what = "full", upper = FALSE)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`neighbourhood` The number of directions in which cell adjacencies are considered as neighbours: 4 (rook's case), 8 (queen's case) or a binary matrix where the ones define the neighbourhood. The default is 4.

`what` Which adjacencies to calculate: "full" for a full adjacency matrix, "like" for the diagonal, "unlike" for the off diagonal part of the matrix and "triangle" for a triangular matrix counting adjacencies only once.

`upper` Logical value indicating whether the upper triangle of the adjacency matrix should be returned (default FALSE).

## Details

A fast implementation with Rcpp to calculate the adjacency matrix for raster. The adjacency matrix is most often used in landscape metrics to describe the configuration of landscapes, is it is a cellwise count of edges between classes.

The "full" adjacency matrix is double-count method, as it contains the pairwise counts of cells between all classes. The diagonal of this matrix contains the like adjacencies, a count for how many edges a shared in each class with the same class.

The "unlike" adjacencies are counting the cellwise edges between different classes.

## Value

matrix with adjacencies between classes in a raster and between cells from the same class.

## Examples

```
# calculate full adjacency matrix
get_adjacencies(landscape, 4)

# count diagonal neighbour adjacencies
diagonal_matrix <- matrix(c(1,  NA,  1,
                            NA,  0, NA,
                            1,  NA,  1), 3, 3, byrow = TRUE)
get_adjacencies(landscape, diagonal_matrix)

# equivalent with the raster package:
adjacencies <- raster::adjacent(landscape, 1:raster::ncell(landscape), 4, pairs=TRUE)
table(landscape[adjacencies[,1]], landscape[adjacencies[,2]])
```

## get_boundaries *get_boundaries*

### Description

Get boundary cells of patches

### Usage

```
get_boundaries(
  landscape,
  consider_boundary,
  edge_depth,
  as_NA,
  patch_id,
  return_raster
)

## S3 method for class 'RasterLayer'
get_boundaries(
  landscape,
  consider_boundary = FALSE,
  edge_depth = 1,
  as_NA = FALSE,
  patch_id = FALSE,
  return_raster = TRUE
)

## S3 method for class 'RasterStack'
get_boundaries(
  landscape,
  consider_boundary = FALSE,
  edge_depth = 1,
  as_NA = FALSE,
  patch_id = FALSE,
  return_raster = TRUE
)

## S3 method for class 'RasterBrick'
get_boundaries(
  landscape,
  consider_boundary = FALSE,
  edge_depth = 1,
  as_NA = FALSE,
  patch_id = FALSE,
  return_raster = TRUE
)
```

```
## S3 method for class 'stars'
get_boundaries(
  landscape,
  consider_boundary = FALSE,
  edge_depth = 1,
  as_NA = FALSE,
  patch_id = FALSE,
  return_raster = TRUE
)

## S3 method for class 'list'
get_boundaries(
  landscape,
  consider_boundary = FALSE,
  edge_depth = 1,
  as_NA = FALSE,
  patch_id = FALSE,
  return_raster = TRUE
)

get_boundaries_calc(landscape, consider_boundary, edge_depth, as_NA, patch_id)
```

### Arguments

| | |
|---|---|
| `landscape` | RasterLayer or matrix. |
| `consider_boundary` | Logical if cells that only neighbour the landscape boundary should be considered as edge. |
| `edge_depth` | Distance (in cells) a cell has the be away from the patch edge to be considered as core cell. |
| `as_NA` | If true, non-boundary cells area labeld NA. |
| `patch_id` | If true, boundary/edge cells are labeled with the original patch id. |
| `return_raster` | If false, matrix is returned. |

### Details

All boundary/edge cells are labeled 1, all non-boundary cells 0. NA values are not changed. Boundary cells are defined as cells that neighbour either a NA cell or a cell with a different value than itself. Non-boundary cells only neighbour cells with the same value than themself.

### Value

List with RasterLayer or matrix

### Examples

```
class_1 <- get_patches(landscape, class = 1)[[1]]
```

```
get_boundaries(class_1)
get_boundaries(class_1, return_raster = FALSE)
```

---

`get_centroids` *get_centroids*

---

### Description

Centroid of patches

### Usage

```
get_centroids(landscape, directions, cell_center, return_sp, verbose)

## S3 method for class 'RasterLayer'
get_centroids(
  landscape,
  directions = 8,
  cell_center = FALSE,
  return_sp = FALSE,
  verbose = TRUE
)

## S3 method for class 'RasterStack'
get_centroids(
  landscape,
  directions = 8,
  cell_center = FALSE,
  return_sp = FALSE,
  verbose = TRUE
)

## S3 method for class 'RasterBrick'
get_centroids(
  landscape,
  directions = 8,
  cell_center = FALSE,
  return_sp = FALSE,
  verbose = TRUE
)

## S3 method for class 'stars'
get_centroids(
  landscape,
  directions = 8,
  cell_center = FALSE,
  return_sp = FALSE,
  verbose = TRUE
)

## S3 method for class 'list'
get_centroids(
  landscape,
  directions = 8,
  cell_center = FALSE,
  return_sp = FALSE,
  verbose = TRUE
)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `cell_center` | If true, the coordinates of the centroid are forced to be a cell center within the patch. |
| `return_sp` | If true, a SpatialPointsDataFrame is returned. |
| `verbose` | Print warning messages |

## Details

Get the coordinates of the centroid of each patch. The centroid is by default defined as the mean location of all cell centers. To force the centroid to be located within each patch, use the `cell_center` argument. In this case, the centroid is defined as the cell center that is the closest to the mean location.

## Examples

```
# get centroid location
get_centroids(landscape)
```

# `get_circumscribingcircle` *get_circumscribingcircle*

## Description

Diameter of the circumscribing circle around patches

## Usage

```
get_circumscribingcircle(landscape, directions, level)

## S3 method for class 'RasterLayer'
get_circumscribingcircle(landscape, directions = 8, level = "patch")

## S3 method for class 'RasterStack'
get_circumscribingcircle(landscape, directions = 8, level = "patch")

## S3 method for class 'RasterBrick'
get_circumscribingcircle(landscape, directions = 8, level = "patch")

## S3 method for class 'stars'
get_circumscribingcircle(landscape, directions = 8, level = "patch")

## S3 method for class 'list'
get_circumscribingcircle(landscape, directions = 8, level = "patch")
```

## Arguments

| | |
|---|---|
| `landscape` | RasterLayer or matrix (with x, y, id columns) |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `level` | Either 'patch' or 'class' for the corresponding level. |

## Details

The diameter of the smallest circumscribing circle around a patch in the landscape is based on the maximum distance between the corners of each cell. This ensures that all cells of the patch are included in the patch.

## References

Based on C++ code from Project Nayuki (https://www.nayuki.io/page/smallest-enclosing-circle).

## Examples

```
# get circle around each patch
get_circumscribingcircle(landscape)

# get circle around whole class
get_circumscribingcircle(landscape, level = "class")
```

## get_nearestneighbour *get_nearestneighbour*

### Description

Euclidean distance to nearest neighbour

### Usage

```
get_nearestneighbour(landscape, return_id)

## S3 method for class 'RasterLayer'
get_nearestneighbour(landscape, return_id = FALSE)

## S3 method for class 'RasterStack'
get_nearestneighbour(landscape, return_id = FALSE)

## S3 method for class 'RasterBrick'
get_nearestneighbour(landscape, return_id = FALSE)

## S3 method for class 'stars'
get_nearestneighbour(landscape, return_id = FALSE)

## S3 method for class 'list'
get_nearestneighbour(landscape, return_id = FALSE)
```

### Arguments

| | |
|---|---|
| `landscape` | RasterLayer or matrix (with x,y,id columns). |
| `return_id` | If TRUE, also the patch ID of the nearest neighbour is returned. |

### Details

Fast and memory safe Rcpp implementation for calculating the minimum Euclidean distances to the nearest patch of the same class in a raster or matrix. All patches need an unique ID (see `get_patches`). Please be aware that the patch ID is not identical to the patch ID of all metric functions (lsm_). If `return_ID = TRUE`, for some focal patches several nearest neighbour patches might be returned.

### References

Based on RCpp code of Florian Privé <`florian.prive.21@gmail.com`>

### Examples

```
# get patches for class 1
class_1 <- get_patches(landscape, class = 2)[[1]]

# calculate the distance between patches
get_nearestneighbour(class_1)
get_nearestneighbour(class_1, return_id = TRUE)
```

## get_patches *get_patches*

### Description

Connected components labeling to derive patches in a landscape.

### Usage

```
get_patches(landscape, class, directions, to_disk, return_raster)

## S3 method for class 'RasterLayer'
get_patches(
  landscape,
  class = "all",
  directions = 8,
  to_disk = getOption("to_disk", default = FALSE),
  return_raster = TRUE
)

## S3 method for class 'RasterStack'
get_patches(
  landscape,
  class = "all",
  directions = 8,
  to_disk = getOption("to_disk", default = FALSE),
  return_raster = TRUE
)

## S3 method for class 'RasterBrick'
get_patches(
  landscape,
  class = "all",
  directions = 8,
  to_disk = getOption("to_disk", default = FALSE),
  return_raster = TRUE
)
```

```
## S3 method for class 'stars'
get_patches(
  landscape,
  class = "all",
  directions = 8,
  to_disk = getOption("to_disk", default = FALSE),
  return_raster = TRUE
)

## S3 method for class 'list'
get_patches(
  landscape,
  class = "all",
  directions = 8,
  to_disk = getOption("to_disk", default = FALSE),
  return_raster = TRUE
)

## S3 method for class 'matrix'
get_patches(
  landscape,
  class = "all",
  directions = 8,
  to_disk = getOption("to_disk", default = FALSE),
  return_raster = FALSE
)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `class` | Either "all" (default) for every class in the raster, or specify class value. See Details. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `to_disk` | Logical argument, if FALSE results of get_patches are hold in memory. If true, get_patches writes temporary files and hence, does not hold everything in memory. Can be set with a global option, e.g. `option(to_disk = TRUE)`. See Details. |
| `return_raster` | If false, matrix is returned |

## Details

Searches for connected patches (neighbouring cells of the same class i). The 8-neighbours rule ('queen's case) or 4-neighbours rule (rook's case) is used. Returns a list with raster. For each class the connected patches have the value 1 - n. All cells not belonging to the class are `NA`.

Landscape metrics rely on the delineation of patches. Hence, `get_patches` is heavily used in **landscapemetrics**. As raster can be quite big, the fact that `get_patches` creates a copy of the raster for each class in a landscape becomes a burden for computer memory. Hence, the argument *to_disk* allows to store the results of the connected labeling algorithm on disk. Furthermore, this option can be set globally, so that every function that internally uses `get_patches` can make use of that.

## Value

List

## References

Vincent, L., Soille, P. 1991. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence. 13 (6), 583-598

## Examples

```
# check for patches of class 1
patched_raster <- get_patches(landscape, class = 1)

# count patches
length(raster::unique(patched_raster[[1]]))

# check for patches of every class
patched_raster <-  get_patches(landscape)
```

# get_unique_values *get_unique_values*

## Description

This function returns the unique values of an object.

## Usage

```
get_unique_values(x, simplify, verbose)

## S3 method for class 'numeric'
get_unique_values(x, simplify = FALSE, verbose = TRUE)

## S3 method for class 'matrix'
get_unique_values(x, simplify = FALSE, verbose = TRUE)

## S3 method for class 'RasterLayer'
get_unique_values(x, simplify = FALSE, verbose = TRUE)

## S3 method for class 'list'
get_unique_values(x, simplify = FALSE, verbose = TRUE)
```

```
## S3 method for class 'RasterStack'
get_unique_values(x, simplify = FALSE, verbose = TRUE)

## S3 method for class 'RasterBrick'
get_unique_values(x, simplify = FALSE, verbose = TRUE)

## S3 method for class 'stars'
get_unique_values(x, simplify = FALSE, verbose = TRUE)
```

## Arguments

| | |
|---|---|
| x | vector, matrix or Raster* object |
| simplify | If true, a vector will be returned instead of a list for 1-dimensional input |
| verbose | If true, warning messages are printend |

## Details

Fast and memory friendly Rcpp implementation to find the unique values of an object.

## Examples

```
get_unique_values(landscape)

landscape_stack <- raster::stack(landscape, landscape, landscape)
get_unique_values(landscape_stack)

landscape_matrix <- raster::as.matrix(landscape)
get_unique_values(landscape_matrix)

x_vec <- c(1, 2, 1, 1, 2, 2)
get_unique_values(x_vec)

landscape_list <- list(landscape, landscape_matrix, x_vec)
get_unique_values(landscape_list)
```

# landscape *Example map (random cluster neutral landscape model).*

## Description

An example map to show landscapetools functionality generated with the `nlm_randomcluster()` algorithm.

## Usage

```
landscape
```

### Format

A raster layer object.

### Source

Simulated neutral landscape model with R. https://github.com/ropensci/NLMR/

## `landscapemetrics` *landscapemetrics*

### Description

Calculates landscape metrics for categorical landscape patterns in a tidy workflow. 'landscapemetrics' reimplements the most common metrics from FRAGSTATS and new ones from the current literature on landscape metrics. This package supports raster spatial objects and takes RasterLayer, RasterStacks, RasterBricks or lists of RasterLayer from the 'raster' package as input arguments. It further provides utility functions to visualize patches, select metrics and building blocks to develop new metrics.

### Author(s)

**Maintainer**: Maximillian H.K. Hesselbarth `<mhk.hesselbarth@gmail.com>` (ORCID)

Authors:

- Marco Sciaini `<sciaini.marco@gmail.com>` (ORCID)
- Jakub Nowosad `<nowosad.jakub@gmail.com>` (ORCID)
- Sebastian Hanss (ORCID)

Other contributors:

- Laura J. Graham (Input on package structure) [contributor]
- Jeffrey Hollister (Input on package structure) [contributor]
- Kimberly A. With (Input on package structure) [contributor]
- Florian Privé (Original author of underlying C++ code for get_nearestneighbour() function) [contributor]
- Project Nayuki (Original author of underlying C++ code for get_circumscribingcircle and lsm_p_circle) [contributor]
- Matt Strimas-Mackey (Bugfix in sample_metrics()) [contributor]

### See Also

Useful links:

- `https://r-spatialecology.github.io/landscapemetrics/`
- Report bugs at `https://github.com/r-spatialecology/landscapemetrics/issues`

## list_lsm *List landscape metrics*

### Description

List landscape metrics

### Usage

```
list_lsm(
  level = NULL,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  simplify = FALSE,
  verbose = TRUE
)
```

### Arguments

| | |
|---|---|
| `level` | Level of metrics. Either 'patch', 'class' or 'landscape' (or vector with combination). |
| `metric` | Abbreviation of metrics (e.g. 'area'). |
| `name` | Full name of metrics (e.g. 'core area') |
| `type` | Type according to FRAGSTATS grouping (e.g. 'aggregation metrics'). |
| `what` | Selected level of metrics: either "patch", "class" or "landscape". It is also possible to specify functions as a vector of strings, e.g. `what = c("lsm_c_ca","lsm_l_ta")`. |
| `simplify` | If true, function names are returned as vector. |
| `verbose` | Print warning messages |

### Details

List all available landscape metrics depending on the provided filter arguments. If an argument is not provided, automatically all possibilities are selected. Therefore, to get **all** available metrics, use simply `list_lsm()`. For all arguments with exception of the `what` argument, it is also possible to use a negative subset, i.e. all metrics **but** the selected ones. Therefore, simply use e.g. `level = "-patch"`. Furthermore, it is possible to only get a vector with all function names instead of the full tibble.

### Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## Examples

```
list_lsm(level = c("patch", "landscape"), type = "aggregation metric")
list_lsm(level = "-patch", type = "area and edge metric")
list_lsm(metric = "area", simplify = TRUE)

list_lsm(metric = "area", what = "lsm_p_shape")
list_lsm(metric = "area", what = c("patch", "lsm_l_ta"))
list_lsm(what = c("lsm_c_tca", "lsm_l_ta"))
```

---

# lsm_abbreviations_names

*Tibble of abbreviations coming from FRAGSTATS*

---

## Description

A single tibble for every abbreviation of every metric that is reimplemented in landscapemetrics and its corresponding full name in the literature.

## Usage

```
lsm_abbreviations_names
```

## Format

A tibble object.

## Details

Can be used after calculating the metric(s) with a join to have a more readable results tibble or for visualizing your results.

## Examples

```
patch_area <- lsm_p_area(landscape)
patch_area <- merge(x = patch_area, y = lsm_abbreviations_names, by = c("level", "metric"))
```

## lsm_c_ai *AI (class level)*

### Description

Aggregation index (Aggregation metric)

### Usage

```
lsm_c_ai(landscape)

## S3 method for class 'RasterLayer'
lsm_c_ai(landscape)

## S3 method for class 'RasterStack'
lsm_c_ai(landscape)

## S3 method for class 'RasterBrick'
lsm_c_ai(landscape)

## S3 method for class 'stars'
lsm_c_ai(landscape)

## S3 method for class 'list'
lsm_c_ai(landscape)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers |

### Details

$$AI = \left[ \frac{g_{ii}}{max - g_{ii}} \right] (100)$$

where $g_{ii}$ is the number of like adjacencies based on the single-count method and $max - g_{ii}$ is the classwise maximum number of like adjacencies of class i.

AI is an 'Aggregation metric'. It equals the number of like adjacencies divided by the theoretical maximum possible number of like adjacencies for that class. The metric is based on he adjacency matrix and the the single-count method.

**Units:** Percent

**Range:** 0 <= AI <= 100

**Behaviour:** Equals 0 for maximally disaggregated and 100 for maximally aggregated classes.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

He, H. S., DeZonia, B. E., & Mladenoff, D. J. 2000. An aggregation index (AI) to quantify spatial patterns of landscapes. Landscape ecology, 15(7), 591-601.

### See Also

lsm_l_ai

### Examples

```
lsm_c_ai(landscape)
```

---

## lsm_c_area_cv *AREA_CV (class level)*

---

### Description

Coefficient of variation of patch area (Area and edge metric)

### Usage

```
lsm_c_area_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_area_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_area_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_area_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_area_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_area_cv(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$AREA_{CV} = cv(AREA[patch_{ij}])$$

where $AREA[patch_{ij}]$ is the area of each patch in hectares.

AREA_CV is an 'Area and Edge metric'. The metric summarises each class as the Coefficient of variation of all patch areas belonging to class i. The metric describes the differences among patches of the same class i in the landscape and is easily comparable because it is scaled to the mean.

**Units:** Hectares

**Range:** AREA_CV >= 0

**Behaviour:** Equals AREA_CV = 0 if all patches are identical in size. Increases, without limit, as the variation of patch areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_area, cv,
lsm_c_area_mn, lsm_c_area_sd,
lsm_l_area_mn, lsm_l_area_sd, lsm_l_area_cv

## Examples

```
lsm_c_area_cv(landscape)
```

## lsm_c_area_mn *AREA_MN (class level)*

### Description

Mean of patch area (Area and edge metric)

### Usage

```
lsm_c_area_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_area_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_area_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_area_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_area_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_area_mn(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$AREA_{MN} = mean(AREA[patch_{ij}])$$

where $AREA[patch_{ij}]$ is the area of each patch in hectares

AREA_MN is an 'Area and Edge metric'. The metric summarises each class as the mean of all patch areas belonging to class i. The metric is a simple way to describe the composition of the landscape. Especially together with the total class area (`lsm_c_ca`), it can also give an an idea of patch structure (e.g. many small patches vs. few larges patches).

**Units:** Hectares

**Range:** AREA_MN > 0

**Behaviour:** Approaches AREA_MN = 0 if all patches are small. Increases, without limit, as the patch areas increase.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_area`, `mean`,
`lsm_c_area_cv`, `lsm_c_area_sd`,
`lsm_l_area_mn`, `lsm_l_area_sd`, `lsm_l_area_cv`

## Examples

```
lsm_c_area_mn(landscape)
```

---

# `lsm_c_area_sd` *AREA_SD (class level)*

---

## Description

Standard deviation of patch area (Area and edge metric)

## Usage

```
lsm_c_area_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_area_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_area_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_area_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_area_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_area_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_division(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$AREA_{SD} = sd(AREA[patch_{ij}])$$

where $AREA[patch_{ij}]$ is the area of each patch in hectares.

AREA_SD is an 'Area and Edge metric'. The metric summarises each class as the standard deviation of all patch areas belonging to class i. The metric describes the differences among patches of the same class i in the landscape.

**Units:** Hectares

**Range:** AREA_SD >= 0

**Behaviour:** Equals AREA_SD = 0 if all patches are identical in size. Increases, without limit, as the variation of patch areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_area`, `sd`,
`lsm_c_area_mn`, `lsm_c_area_cv`,
`lsm_l_area_mn`, `lsm_l_area_sd`, `lsm_l_area_cv`

## Examples

```
lsm_c_area_sd(landscape)
```

## lsm_c_ca *CA (class level)*

### Description

Total (class) area (Area and edge metric)

### Usage

```
lsm_c_ca(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_ca(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_ca(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_ca(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_ca(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_ca(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$CA = sum(AREA[patch_{ij}])$$

where $AREA[patch_{ij}]$ is the area of each patch in hectares.

CA is an 'Area and edge metric' and a measure of composition. The total (class) area sums the area of all patches belonging to class i. It shows if the landscape is e.g. dominated by one class or if all classes are equally present. CA is an absolute measure, making comparisons among landscapes with different total areas difficult.

**Units:** Hectares

**Range:** CA > 0

**Behaviour:** Approaches CA > 0 as the patch areas of class i become small. Increases, without limit, as the patch areas of class i become large. CA = TA if only one class is present.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

### See Also

lsm_p_area, sum,
lsm_l_ta

### Examples

```
lsm_c_ca(landscape)
```

## lsm_c_cai_cv *CAI_CV (class level)*

### Description

Coefficient of variation of core area index (Core area metric)

### Usage

```
lsm_c_cai_cv(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_cai_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_c_cai_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
```

```
lsm_c_cai_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_c_cai_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_c_cai_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `consider_boundary` | Logical if cells that only neighbour the landscape boundary should be considered as core |
| `edge_depth` | Distance (in cells) a cell has the be away from the patch edge to be considered as core cell |

## Details

$$CAI_{CV} = cv(CAI[patch_{ij}]$$

where $CAI[patch_{ij}]$ is the core area index of each patch.

CAI_CV is a 'Core area metric'. The metric summarises each class as the Coefficient of variation of the core area index of all patches belonging to class i. The core area index is the percentage of core area in relation to patch area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). The metric describes the differences among patches of the same class i in the landscape. Because it is scaled to the mean, it is easily comparable.

**Units:** Percent

**Range:** CAI_CV >= 0

**Behaviour:** Equals CAI_CV = 0 if the core area index is identical for all patches. Increases, without limit, as the variation of the core area indices increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

### See Also

lsm_p_cai, cv,
lsm_c_cai_mn, lsm_c_cai_sd,
lsm_l_cai_mn, lsm_l_cai_sd, lsm_l_cai_cv

### Examples

```
lsm_c_cai_cv(landscape)
```

## lsm_c_cai_mn *CAI_MN (class level)*

### Description

Mean of core area index (Core area metric)

### Usage

```
lsm_c_cai_mn(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_cai_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_c_cai_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_c_cai_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_c_cai_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_c_cai_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| consider_boundary | Logical if cells that only neighbour the landscape boundary should be considered as core |
| edge_depth | Distance (in cells) a cell has the be away from the patch edge to be considered as core cell |

## Details

$$CAI_{MN} = mean(CAI[patch_{ij}]$$

where $CAI[patch_{ij}]$ is the core area index of each patch.

CAI_MN is a 'Core area metric'. The metric summarises each class as the mean of the core area index of all patches belonging to class i. The core area index is the percentage of core area in relation to patch area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case).

**Units:** Percent

**Range:** 0 <= CAI_MN <= 100

**Behaviour:** CAI_MN = 0 when all patches have no core area and approaches CAI_MN = 100 with increasing percentage of core area within patches.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_cai`, `mean`,
`lsm_c_cai_sd`, `lsm_c_cai_cv`,
`lsm_l_cai_mn`, `lsm_l_cai_sd`, `lsm_l_cai_cv`

## Examples

```
lsm_c_cai_mn(landscape)
```

---

# `lsm_c_cai_sd` *CAI_SD (class level)*

---

## Description

Standard deviation of core area index (Core area metric)

## Usage

```
lsm_c_cai_sd(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_cai_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
```

```
lsm_c_cai_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_c_cai_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_c_cai_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_c_cai_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `consider_boundary` | Logical if cells that only neighbour the landscape boundary should be considered as core |
| `edge_depth` | Distance (in cells) a cell has the be away from the patch edge to be considered as core cell |

## Details

$$CAI_{SD} = sd(CAI[patch_{ij}]$$

where $CAI[patch_{ij}]$ is the core area index of each patch.

CAI_SD is a 'Core area metric'. The metric summarises each class as the standard deviation of the core area index of all patches belonging to class i. The core area index is the percentage of core area in relation to patch area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). The metric describes the differences among patches of the same class i in the landscape.

**Units:** Percent

**Range:** CAI_SD >= 0

**Behaviour:** Equals CAI_SD = 0 if the core area index is identical for all patches. Increases, without limit, as the variation of core area indices increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_cai, sd,
lsm_c_cai_mn, lsm_c_cai_cv,
lsm_l_cai_mn, lsm_l_cai_sd, lsm_l_cai_cv

## Examples

```
lsm_c_cai_sd(landscape)
```

---

# lsm_c_circle_cv *CIRCLE_CV (Class level)*

---

## Description

Coefficient of variation of related circumscribing circle (Shape metric)

## Usage

```
lsm_c_circle_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_circle_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_circle_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_circle_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_circle_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_circle_cv(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$CIRCLE_{CV} = cv(CIRCLE[patch_{ij}])$$

where $CIRCLE[patch_{ij}]$ is the related circumscribing circle of each patch.

CIRCLE_CV is a 'Shape metric' and summarises each class as the Coefficient of variation of the related circumscribing circle of all patches belonging to class i. CIRCLE describes the ratio between the patch area and the smallest circumscribing circle of the patch and characterises the compactness of the patch. CIRCLE_CV describes the differences among patches of the same class i in the landscape. Because it is scaled to the mean, it is easily comparable.

**Units:** None

**Range:** CIRCLE_CV >= 0

**Behaviour:** Equals CIRCLE_CV if the related circumscribing circle is identical for all patches. Increases, without limit, as the variation of related circumscribing circles increases.

## Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Baker, W. L., and Y. Cai. 1992. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. Landscape Ecology 7: 291-302.

Based on C++ code from Project Nayuki (https://www.nayuki.io/page/smallest-enclosing-circle).

### See Also

lsm_p_circle, mean,
lsm_c_circle_mn, lsm_c_circle_sd,
lsm_l_circle_mn, lsm_l_circle_sd, lsm_l_circle_cv

### Examples

```
lsm_c_circle_cv(landscape)
```

## lsm_c_circle_mn *CIRCLE_MN (Class level)*

### Description

Mean of related circumscribing circle (Shape metric)

### Usage

```
lsm_c_circle_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_circle_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_circle_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_circle_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_circle_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_circle_mn(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$CIRCLE_{MN} = mean(CIRCLE[patch_{ij}])$$

where $CIRCLE[patch_{ij}]$ is the related circumscribing circle of each patch.

CIRCLE_MN is a 'Shape metric' and summarises each class as the mean of the related circumscribing circle of all patches belonging to class i. CIRCLE describes the ratio between the patch area and the smallest circumscribing circle of the patch and characterises the compactness of the patch.

**Units:** None

**Range:** CIRCLE_MN > 0

**Behaviour:** Approaches CIRCLE_MN = 0 if the related circumscribing circle of all patches is small. Increases, without limit, as the related circumscribing circles increase.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Baker, W. L., and Y. Cai. 1992. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. Landscape Ecology 7: 291-302.

Based on C++ code from Project Nayuki (https://www.nayuki.io/page/smallest-enclosing-circle).

## See Also

`lsm_p_circle`, `mean`,
`lsm_c_circle_sd`, `lsm_c_circle_cv`,
`lsm_l_circle_mn`, `lsm_l_circle_sd`, `lsm_l_circle_cv`

## Examples

```
lsm_c_circle_mn(landscape)
```

## lsm_c_circle_sd *CIRCLE_SD (Class level)*

### Description

Standard deviation of related circumscribing circle (Shape metric)

### Usage

```
lsm_c_circle_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_circle_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_circle_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_circle_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_circle_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_circle_sd(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$CIRCLE_{SD} = sd(CIRCLE[patch_{ij}])$$

where $CIRCLE[patch_{ij}]$ is the related circumscribing circle of each patch.

CIRCLE_SD is a 'Shape metric' and summarises each class as the standard deviation of the related circumscribing circle of all patches belonging to class i. CIRCLE describes the ratio between the patch area and the smallest circumscribing circle of the patch and characterises the compactness of the patch. The metric describes the differences among patches of the same class i in the landscape.

**Units:** None

**Range:** CIRCLE_SD >= 0

**Behaviour:** Equals CIRCLE_SD if the related circumscribing circle is identical for all patches. Increases, without limit, as the variation of related circumscribing circles increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Baker, W. L., and Y. Cai. 1992. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. Landscape Ecology 7: 291-302.

Based on C++ code from Project Nayuki (https://www.nayuki.io/page/smallest-enclosing-circle).

### See Also

`lsm_p_circle`, `mean`,
`lsm_c_circle_mn`, `lsm_c_circle_cv`,
`lsm_l_circle_mn`, `lsm_l_circle_sd`, `lsm_l_circle_cv`

### Examples

```
lsm_c_circle_sd(landscape)
```

---

## lsm_c_clumpy *CLUMPY (class level)*

---

### Description

Clumpiness index (Aggregation metric)

### Usage

```
lsm_c_clumpy(landscape)

## S3 method for class 'RasterLayer'
lsm_c_clumpy(landscape)

## S3 method for class 'RasterStack'
lsm_c_clumpy(landscape)

## S3 method for class 'RasterBrick'
lsm_c_clumpy(landscape)

## S3 method for class 'stars'
lsm_c_clumpy(landscape)

## S3 method for class 'list'
lsm_c_clumpy(landscape)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers |

## Details

$$Given G_i = \left( \frac{g_{ii}}{(\sum\limits_{k=1}^{m} g_{ik}) - mine_i} \right)$$

$$CLUMPY = \left[ \frac{G_i - P_i}{P_i} for G_i < P_i \& P_i < .5; else \frac{G_i - P_i}{1 - P_i} \right]$$

where $g_{ii}$ is the number of like adjacencies, $g_{ik}$ is the classwise number of all adjacencies including the focal class, $mine_i$ is the minimum perimeter of the total class in terms of cell surfaces assuming total clumping and $P_i$ is the proportion of landscape occupied by each class.

CLUMPY is an 'Aggregation metric'. It equals the proportional deviation of the proportion of like adjacencies involving the corresponding class from that expected under a spatially random distribution. The metric is based on he adjacency matrix and the the double-count method.

**Units:** None

, directions = directions

**Range:** -1 <= CLUMPY <= 1

**Behaviour:** Equals -1 for maximally disaggregated, 0 for randomly distributed and 1 for maximally aggregated classes.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## Examples

```
lsm_c_clumpy(landscape)
```

## `lsm_c_cohesion` *COHESION (class level)*

### Description

Patch Cohesion Index (Aggregation metric)

### Usage

```
lsm_c_cohesion(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_cohesion(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_cohesion(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_cohesion(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_cohesion(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_cohesion(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$COHESION = 1 - (\frac{\sum\limits_{j=1}^{n} p_{ij}}{\sum\limits_{j=1}^{n} p_{ij} \sqrt{a_{ij}}}) * (1 - \frac{1}{\sqrt{Z}})^{-1} * 100$$

where $p_{ij}$ is the perimeter in meters, $a_{ij}$ is the area in square meters and $Z$ is the number of cells.

COHESION is an 'Aggregation metric'. It characterises the connectedness of patches belonging to class i. It can be used to asses if patches of the same class are located aggregated or rather isolated and thereby COHESION gives information about the configuration of the landscape.

**Units:** Percent

**Ranges:** 0 < COHESION < 100

**Behaviour:** Approaches COHESION = 0 if patches of class i become more isolated. Increases if patches of class i become more aggregated.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Schumaker, N. H. 1996. Using landscape indices to predict habitat connectivity. Ecology, 77(4), 1210-1225.

## See Also

lsm_p_perim, lsm_p_area,
lsm_l_cohesion

## Examples

```
lsm_c_cohesion(landscape)
```

---

# lsm_c_contig_cv *CONTIG_CV (class level)*

---

## Description

Coefficient of variation of Contiguity index (Shape metric)

## Usage

```
lsm_c_contig_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_contig_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_contig_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_contig_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_contig_cv(landscape, directions = 8)
```

```
## S3 method for class 'list'
lsm_c_contig_cv(landscape, directions = 8)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

## Details

$$CONTIG_{CV} = cv(CONTIG[patch_{ij}])$$

where $CONTIG[patch_{ij}]$ is the contiguity of each patch.

CONTIG_CV is a 'Shape metric'. It summarises each class as the mean of each patch belonging to class i. CONTIG_CV asses the spatial connectedness (contiguity) of cells in patches. The metric coerces patch values to a value of 1 and the background to NA. A nine cell focal filter matrix:

```
filter_matrix <- matrix(c(1, 2, 1,
                          2, 1, 2,
                          1, 2, 1), 3, 3, byrow = T)
```

... is then used to weight orthogonally contiguous pixels more heavily than diagonally contiguous pixels. Therefore, larger and more connections between patch cells in the rookie case result in larger contiguity index values.

**Units:** None

**Range:** CONTIG_CV >= 0

**Behaviour:** CONTIG_CV = 0 if the contiguity index is identical for all patches. Increases, without limit, as the variation of CONTIG increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

LaGro, J. 1991. Assessing patch shape in landscape mosaics. Photogrammetric Engineering and Remote Sensing, 57(3), 285-293

## See Also

`lsm_p_contig`, `lsm_c_contig_mn`, `lsm_c_contig_cv`,
`lsm_l_contig_mn`, `lsm_l_contig_sd`, `lsm_l_contig_cv`

## Examples

```
lsm_c_contig_cv(landscape)
```

---

`lsm_c_contig_mn` *CONTIG_MN (class level)*

---

## Description

Mean of Contiguity index (Shape metric)

## Usage

```
lsm_c_contig_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_contig_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_contig_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_contig_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_contig_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_contig_mn(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$CONTIG_{MN} = mean(CONTIG[patch_{ij}])$$

where $CONTIG[patch_{ij}]$ is the contiguity of each patch.

CONTIG_MN is a 'Shape metric'. It summarises each class as the mean of each patch belonging to class i. CONTIG_MN asses the spatial connectedness (contiguity) of cells in patches. The metric coerces patch values to a value of 1 and the background to NA. A nine cell focal filter matrix:

```
filter_matrix <- matrix(c(1, 2, 1,
                          2, 1, 2,
                          1, 2, 1), 3, 3, byrow = T)
```

... is then used to weight orthogonally contiguous pixels more heavily than diagonally contiguous pixels. Therefore, larger and more connections between patch cells in the rookie case result in larger contiguity index values.

**Units:** None

**Range:** 0 >= CONTIG_MN <= 1

**Behaviour:** CONTIG equals the mean of the contiguity index on class level for all patches.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

LaGro, J. 1991. Assessing patch shape in landscape mosaics. Photogrammetric Engineering and Remote Sensing, 57(3), 285-293

## See Also

lsm_p_contig, lsm_c_contig_sd, lsm_c_contig_cv,
lsm_l_contig_mn, lsm_l_contig_sd, lsm_l_contig_cv

## Examples

```
lsm_c_contig_mn(landscape)
```

# lsm_c_contig_sd *CONTIG_SD (class level)*

## Description

Standard deviation of Contiguity index (Shape metric)

## Usage

```
lsm_c_contig_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_contig_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_contig_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_contig_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_contig_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_contig_sd(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$CONTIG_{SD} = sd(CONTIG[patch_{ij}])$$

where $CONTIG[patch_{ij}]$ is the contiguity of each patch.

CONTIG_SD is a 'Shape metric'. It summarises each class as the mean of each patch belonging to class i. CONTIG_SD asses the spatial connectedness (contiguity) of cells in patches. The metric coerces patch values to a value of 1 and the background to NA. A nine cell focal filter matrix:

```
filter_matrix <- matrix(c(1, 2, 1,
                          2, 1, 2,
                          1, 2, 1), 3, 3, byrow = T)
```

... is then used to weight orthogonally contiguous pixels more heavily than diagonally contiguous pixels. Therefore, larger and more connections between patch cells in the rookie case result in larger contiguity index values.

**Units:** None

**Range:** CONTIG_CV >= 0

**Behaviour:** CONTIG_SD = 0 if the contiguity index is identical for all patches. Increases, without limit, as the variation of CONTIG increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

LaGro, J. 1991. Assessing patch shape in landscape mosaics. Photogrammetric Engineering and Remote Sensing, 57(3), 285-293

### See Also

lsm_p_contig, lsm_c_contig_mn, lsm_c_contig_cv,
lsm_l_contig_mn, lsm_l_contig_sd, lsm_l_contig_cv

### Examples

```
lsm_c_contig_sd(landscape)
```

---

## lsm_c_core_cv *CORE_CV (class level)*

---

### Description

Coefficient of variation of core area (Core area metric)

### Usage

```
lsm_c_core_cv(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_core_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_c_core_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

```
## S3 method for class 'RasterBrick'
lsm_c_core_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_c_core_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_c_core_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$CORE_{CV} = cv(CORE[patch_{ij}])$$

where $CORE[patch_{ij}]$ is the core area in square meters of each patch.

CORE_CV is a 'Core area metric'. It equals the Coefficient of variation of the core area of each patch belonging to class i. The core area is defined as all cells that have no neighbour with a different value than themselves (rook's case). The metric describes the differences among patches of the same class i in the landscape and is easily comparable because it is scaled to the mean.

**Units:** Hectares

**Range:** CORE_CV >= 0

**Behaviour:** Equals CORE_CV = 0 if all patches have the same core area. Increases, without limit, as the variation of patch core areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_core, cv,
lsm_c_core_mn, lsm_c_core_sd,
lsm_l_core_mn, lsm_l_core_sd, lsm_l_core_cv

## Examples

```
lsm_c_core_cv(landscape)
```

---

# lsm_c_core_mn *CORE_MN (class level)*

---

## Description

Mean of core area (Core area metric)

## Usage

```
lsm_c_core_mn(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_core_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_c_core_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_c_core_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_c_core_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_c_core_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$CORE_{MN} = mean(CORE[patch_{ij}])$$

where $CORE[patch_{ij}]$ is the core area in square meters of each patch.

CORE_MN is a 'Core area metric' and equals the mean of core areas of all patches belonging to class i. The core area is defined as all cells that have no neighbour with a different value than themselves (rook's case).

**Units:** Hectares

**Range:** CORE_MN >= 0

**Behaviour:** Equals CORE_MN = 0 if CORE = 0 for all patches. Increases, without limit, as the core area indices increase.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_core`, `mean`,
`lsm_c_core_sd`, `lsm_c_core_cv`,
`lsm_l_core_mn`, `lsm_l_core_sd`, `lsm_l_core_cv`

## Examples

```
lsm_c_core_mn(landscape)
```

---

# `lsm_c_core_sd` *CORE_SD (class level)*

---

## Description

Standard deviation patch core area (class level)

## Usage

```
lsm_c_core_sd(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_core_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
```

```
lsm_c_core_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_c_core_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_c_core_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_c_core_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `consider_boundary` | Logical if cells that only neighbour the landscape boundary should be considered as core |
| `edge_depth` | Distance (in cells) a cell has the be away from the patch edge to be considered as core cell |

## Details

$$CORE_{SD} = sd(CORE[patch_{ij}])$$

where $CORE[patch_{ij}]$ is the core area in square meters of each patch.

CORE_SD is a 'Core area metric'. It equals the standard deviation of the core area of each patch belonging to class i. The core area is defined as all cells that have no neighbour with a different value than themselves (rook's case). The metric describes the differences among patches of the same class i in the landscape.

**Units:** Hectares

**Range:** CORE_SD >= 0

**Behaviour:** Equals CORE_SD = 0 if all patches have the same core area. Increases, without limit, as the variation of patch core areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_core, sd,
lsm_c_core_mn, lsm_c_core_cv,
lsm_l_core_mn, lsm_l_core_sd, lsm_l_core_cv

## Examples

```
lsm_c_core_sd(landscape)
```

# lsm_c_cpland *CPLAND (class level)*

## Description

Core area percentage of landscape (Core area metric)

## Usage

```
lsm_c_cpland(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_cpland(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_c_cpland(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_c_cpland(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_c_cpland(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_c_cpland(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$CPLAND = (\frac{\sum_{j=1}^{n} a_{ij}^{core}}{A}) * 100$$

where $a_{ij}^{core}$ is the core area in square meters and $A$ is the total landscape area in square meters.

CPLAND is a 'Core area metric'. It is the percentage of core area of class i in relation to the total landscape area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). Because CPLAND is a relative measure, it is comparable among landscapes with different total areas.

**Units:** Percentage

**Range:** 0 <= CPLAND < 100

**Behaviour:** Approaches CPLAND = 0 if CORE = 0 for all patches. Increases as the amount of core area increases, i.e. patches become larger while being rather simple in shape.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_core` and `lsm_l_ta`

## Examples

```
lsm_c_cpland(landscape)
```

---

# `lsm_c_dcad` *DCAD (class level)*

---

## Description

Disjunct core area density (core area metric)

## Usage

```
lsm_c_dcad(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_dcad(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_c_dcad(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_c_dcad(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_c_dcad(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_c_dcad(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

`consider_boundary`
: Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth`
: Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$DCAD = (\frac{\sum_{j=1}^{n} n_{ij}^{core}}{A}) * 10000 * 100$$

where $n_{ij}^{core}$ is the number of disjunct core areas and $A$ is the total landscape area in square meters.

DCAD is a 'Core area metric'. It equals the number of disjunct core areas per 100 ha relative to the total area. A disjunct core area is a 'patch within the patch' containing only core cells. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). The metric is relative and therefore comparable among landscapes with different total areas.

**Units:** Number per 100 hectares

**Range:** DCAD >= 0

**Behaviour:** Equals DCAD = 0 when DCORE = 0, i.e. no patch of class i contains a disjunct core area. Increases, without limit, as disjunct core areas become more present, i.e. patches becoming larger and less complex.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_c_ndca`, `lsm_l_ta`,
`lsm_l_dcad`

## Examples

```
lsm_c_dcad(landscape)
```

## lsm_c_dcore_cv *DCORE_CV (class level)*

### Description

Coefficient of variation number of disjunct core areas (Core area metric)

### Usage

```
lsm_c_dcore_cv(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_dcore_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_c_dcore_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_c_dcore_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_c_dcore_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_c_dcore_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$DCORE_{CV} = cv(NCORE[patch_{ij}])$$

where $NCORE[patch_{ij}]$ is the number of core areas.

DCORE_CV is an 'Core area metric'. It summarises each class as the Coefficient of variation of all patch areas belonging to class i. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). NCORE counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells. The metric describes the differences among patches of the same class i in the landscape and is easily comparable because it is scaled to the mean.

**Units:** None

**Range:** DCORE_CV >= 0

**Behaviour:** Equals DCORE_CV = 0 if all patches have the same number of disjunct core areas. Increases, without limit, as the variation of number of disjunct core areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_ncore`, `cv`,
`lsm_c_dcore_mn`, `lsm_c_dcore_sd`,
`lsm_l_dcore_mn`, `lsm_l_dcore_sd`, `lsm_l_dcore_cv`

## Examples

```
lsm_c_dcore_cv(landscape)
```

## lsm_c_dcore_mn *DCORE_MN (class level)*

### Description

Mean number of disjunct core areas (Core area metric)

### Usage

```
lsm_c_dcore_mn(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_dcore_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_c_dcore_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_c_dcore_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_c_dcore_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

```
## S3 method for class 'list'
lsm_c_dcore_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$DCORE_{MN} = mean(NCORE[patch_{ij}])$$

where $NCORE[patch_{ij}]$ is the number of core areas.

DCORE_MN is an 'Core area metric'. It summarises each class as the mean of all patch areas belonging to class i. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). NCORE counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells.

**Units:** None

**Range:** DCORE_MN > 0

**Behaviour:** Equals DCORE_MN = 0 if NCORE = 0 for all patches. Increases, without limit, as the number of disjunct core areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_ncore, mean,
lsm_c_dcore_sd, lsm_c_dcore_cv,
lsm_l_dcore_mn, lsm_l_dcore_sd, lsm_l_dcore_cv

## Examples

```
lsm_c_dcore_mn(landscape)
```

---

# lsm_c_dcore_sd *DCORE_SD (class level)*

---

## Description

Standard deviation number of disjunct core areas (Core area metric)

## Usage

```
lsm_c_dcore_sd(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_dcore_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_c_dcore_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_c_dcore_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_c_dcore_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_c_dcore_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$DCORE_{SD} = sd(NCORE[patch_{ij}])$$

where $NCORE[patch_{ij}]$ is the number of core areas.

DCORE_SD is an 'Core area metric'. It summarises each class as the standard deviation of all patch areas belonging to class i. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). NCORE counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells. The metric describes the differences among patches of the same class i in the landscape.

**Units:** None

**Range:** DCORE_SD >= 0

**Behaviour:** Equals DCORE_SD = 0 if all patches have the same number of disjunct core areas. Increases, without limit, as the variation of number of disjunct core areas increases.

## Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

### See Also

`lsm_p_ncore`, `sd`,
`lsm_c_dcore_mn`, `lsm_c_dcore_cv`,
`lsm_l_dcore_mn`, `lsm_l_dcore_sd`, `lsm_l_dcore_cv`

### Examples

```
lsm_c_dcore_sd(landscape)
```

---

## `lsm_c_division` *DIVISION (class level)*

---

### Description

Landscape division index (Aggregation metric)

### Usage

```
lsm_c_division(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_division(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_division(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_division(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_division(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$DIVISON = (1 - \sum_{j=1}^{n} (\frac{a_{ij}}{A})^2)$$

where $a_{ij}$ is the area in square meters and $A$ is the total landscape area in square meters.

DIVISION is an 'Aggregation metric. It can be in as the probability that two randomly selected cells are not located in the same patch of class i. The landscape division index is negatively correlated with the effective mesh size (`lsm_c_mesh`).

**Units:** Proportion

**Ranges:** 0 <= Division < 1

**Behaviour:** Equals DIVISION = 0 if only one patch is present. Approaches DIVISION = 1 if all patches of class i are single cells.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Jaeger, J. A. 2000. Landscape division, splitting index, and effective mesh size: new measures of landscape fragmentation. Landscape ecology, 15(2), 115-130.

## See Also

`lsm_p_area`, `lsm_l_ta`,
`lsm_l_division`

## Examples

```
lsm_c_division(landscape)
```

## lsm_c_ed *ED (class level)*

### Description

Edge Density (Area and Edge metric)

### Usage

```
lsm_c_ed(landscape, count_boundary, directions)

## S3 method for class 'RasterLayer'
lsm_c_ed(landscape, count_boundary = FALSE, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_ed(landscape, count_boundary = FALSE, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_ed(landscape, count_boundary = FALSE, directions = 8)

## S3 method for class 'stars'
lsm_c_ed(landscape, count_boundary = FALSE, directions = 8)

## S3 method for class 'list'
lsm_c_ed(landscape, count_boundary = FALSE, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `count_boundary` | Count landscape boundary as edge. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$ED = \frac{\sum\limits_{k=1}^{m} e_{ik}}{A} * 10000$$

where $e_{ik}$ is the total edge length in meters and $A$ is the total landscape area in square meters.

ED is an 'Area and Edge metric'. The edge density equals the sum of all edges of class i in relation to the landscape area. The boundary of the landscape is only included in the corresponding total class edge length if `count_boundary = TRUE`. The metric describes the configuration of the landscape, e.g. because an aggregation of the same class will result in a low edge density. The metric is standardized to the total landscape area, and therefore comparisons among landscapes with different total areas are possible.

**Units:** Meters per hectare

**Range:** ED >= 0

**Behaviour:** Equals ED = 0 if only one patch is present (and the landscape boundary is not included) and increases, without limit, as the landscapes becomes more patchy

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_c_te`, `lsm_l_ta`,
`lsm_l_ed`

## Examples

```
lsm_c_ed(landscape)
```

# `lsm_c_enn_cv` *ENN_CV (class level)*

## Description

Coefficient of variation of euclidean nearest-neighbor distance (Aggregation metric)

## Usage

```
lsm_c_enn_cv(landscape, directions, verbose)

## S3 method for class 'RasterLayer'
lsm_c_enn_cv(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_c_enn_cv(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_c_enn_cv(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'stars'
```

```
lsm_c_enn_cv(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_c_enn_cv(landscape, directions = 8, verbose = TRUE)
```

## Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| verbose | Print warning message if not sufficient patches are present |

## Details

$$ENN_{CV} = cv(ENN[patch_{ij}])$$

where $ENN[patch_{ij}]$ is the euclidean nearest-neighbor distance of each patch.

ENN_CV is an 'Aggregation metric'. It summarises each class as the Coefficient of variation of each patch belonging to class i. ENN measures the distance to the nearest neighbouring patch of the same class i. The distance is measured from edge-to-edge. The range is limited by the cell resolution on the lower limit and the landscape extent on the upper limit. The metric is a simple way to describe patch isolation. Because it is scaled to the mean, it is easily comparable among different landscapes.

**Units:** Meters

**Range:** ENN_CV >= 0

**Behaviour:** Equals ENN_CV = 0 if the euclidean nearest-neighbor distance is identical for all patches. Increases, without limit, as the variation of ENN increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

McGarigal, K., and McComb, W. C. (1995). Relationships between landscape structure and breeding birds in the Oregon Coast Range. Ecological monographs, 65(3), 235-260.

## See Also

lsm_p_enn, cv,
lsm_c_enn_mn, lsm_c_enn_sd,
lsm_l_enn_mn, lsm_l_enn_sd, lsm_l_enn_cv

## Examples

```
lsm_c_enn_cv(landscape)
```

---

`lsm_c_enn_mn` *ENN_MN (class level)*

---

## Description

Mean of euclidean nearest-neighbor distance (Aggregation metric)

## Usage

```
lsm_c_enn_mn(landscape, directions, verbose)

## S3 method for class 'RasterLayer'
lsm_c_enn_mn(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_c_enn_mn(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_c_enn_mn(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'stars'
lsm_c_enn_mn(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_c_enn_mn(landscape, directions = 8, verbose = TRUE)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `verbose` | Print warning message if not sufficient patches are present |

## Details

$$ENN_{MN} = mean(ENN[patch_{ij}])$$

where $ENN[patch_{ij}]$ is the euclidean nearest-neighbor distance of each patch.

ENN_MN is an 'Aggregation metric'. It summarises each class as the mean of each patch belonging to class i. ENN measures the distance to the nearest neighbouring patch of the same class i. The distance is measured from edge-to-edge. The range is limited by the cell resolution on the lower limit and the landscape extent on the upper limit.

**Units:** Meters

**Range:** ENN_MN > 0

**Behaviour:** Approaches ENN_MN = 0 as the distance to the nearest neighbour decreases, i.e. patches of the same class i are more aggregated. Increases, without limit, as the distance between neighbouring patches of the same class i increases, i.e. patches are more isolated.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

McGarigal, K., and McComb, W. C. (1995). Relationships between landscape structure and breeding birds in the Oregon Coast Range. Ecological monographs, 65(3), 235-260.

## See Also

lsm_p_enn, mean,
lsm_c_enn_sd, lsm_c_enn_cv,
lsm_l_enn_mn, lsm_l_enn_sd, lsm_l_enn_cv

## Examples

```
lsm_c_enn_mn(landscape)
```

# lsm_c_enn_sd *ENN_SD (class level)*

## Description

Standard deviation of euclidean nearest-neighbor distance (Aggregation metric)

## Usage

```
lsm_c_enn_sd(landscape, directions, verbose)

## S3 method for class 'RasterLayer'
lsm_c_enn_sd(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_c_enn_sd(landscape, directions = 8, verbose = TRUE)
```

```
## S3 method for class 'RasterBrick'
lsm_c_enn_sd(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'stars'
lsm_c_enn_sd(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_c_enn_sd(landscape, directions = 8, verbose = TRUE)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `verbose` | Print warning message if not sufficient patches are present |

## Details

$$ENN_{SD} = sd(ENN[patch_{ij}])$$

where $ENN[patch_{ij}]$ is the euclidean nearest-neighbor distance of each patch.

ENN_CV is an 'Aggregation metric'. It summarises each class as the standard deviation of each patch belonging to class i. ENN measures the distance to the nearest neighbouring patch of the same class i. The distance is measured from edge-to-edge. The range is limited by the cell resolution on the lower limit and the landscape extent on the upper limit. The metric is a simple way to describe patch isolation. Because it is scaled to the mean, it is easily comparable among different landscapes.

**Units:** Meters

**Range:** ENN_SD >= 0

**Behaviour:** Equals ENN_SD = 0 if the euclidean nearest-neighbor distance is identical for all patches. Increases, without limit, as the variation of ENN increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

McGarigal, K., and McComb, W. C. (1995). Relationships between landscape structure and breeding birds in the Oregon Coast Range. Ecological monographs, 65(3), 235-260.

### See Also

lsm_p_enn, sd,
lsm_c_enn_mn, lsm_c_enn_cv,
lsm_l_enn_mn, lsm_l_enn_sd, lsm_l_enn_cv

### Examples

```
lsm_c_enn_sd(landscape)
```

## lsm_c_frac_cv *FRAC_CV (class level)*

### Description

Coefficient of variation fractal dimension index (Shape metric)

### Usage

```
lsm_c_frac_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_frac_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_frac_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_frac_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_frac_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_frac_cv(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$FRAC_{CV} = cv(FRAC[patch_{ij}])$$

where $FRAC[patch_{ij}]$ equals the fractal dimension index of each patch.

FRAC_CV is a 'Shape metric'. The metric summarises each class as the Coefficient of variation of the fractal dimension index of all patches belonging to class i. The fractal dimension index is based on the patch perimeter and the patch area and describes the patch complexity. The Coefficient of variation is scaled to the mean and comparable among different landscapes.

**Units:** None

**Range:** FRAC_CV >= 0

**Behaviour:** Equals FRAC_CV = 0 if the fractal dimension index is identical for all patches. Increases, without limit, as the variation of the fractal dimension indices increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Mandelbrot, B. B. 1977. Fractals: Form, Chance, and Dimension. San Francisco. W. H. Freeman and Company.

## See Also

lsm_p_frac, cv,
lsm_c_frac_mn, lsm_c_frac_sd,
lsm_l_frac_mn, lsm_l_frac_sd, lsm_l_frac_cv

## Examples

```
lsm_c_frac_cv(landscape)
```

## lsm_c_frac_mn *FRAC_MN (class level)*

### Description

Mean fractal dimension index (Shape metric)

### Usage

```
lsm_c_frac_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_frac_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_frac_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_frac_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_frac_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_frac_mn(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$FRAC_{MN} = mean(FRAC[patch_{ij}])$$

where $FRAC[patch_{ij}]$ equals the fractal dimension index of each patch.

FRAC_MN is a 'Shape metric'. The metric summarises each class as the mean of the fractal dimension index of all patches belonging to class i. The fractal dimension index is based on the patch perimeter and the patch area and describes the patch complexity. The Coefficient of variation is scaled to the mean and comparable among different landscapes.

**Units:** None

**Range:** FRAC_MN > 0

**Behaviour:** Approaches FRAC_MN = 1 if all patches are squared and FRAC_MN = 2 if all patches are irregular.

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Mandelbrot, B. B. 1977. Fractals: Form, Chance, and Dimension. San Francisco. W. H. Freeman and Company.

### See Also

`lsm_p_frac`, `mean`,
`lsm_c_frac_sd`, `lsm_c_frac_cv`,
`lsm_l_frac_mn`, `lsm_l_frac_sd`, `lsm_l_frac_cv`

### Examples

```
lsm_c_frac_mn(landscape)
```

---

## lsm_c_frac_sd *FRAC_SD (class level)*

---

### Description

Standard deviation fractal dimension index (Shape metric)

### Usage

```
lsm_c_frac_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_frac_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_frac_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_frac_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_frac_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_frac_sd(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$FRAC_{SD} = sd(FRAC[patch_{ij}])$$

where $FRAC[patch_{ij}]$ equals the fractal dimension index of each patch.

FRAC_SD is a 'Shape metric'. The metric summarises each class as the standard deviation of the fractal dimension index of all patches belonging to class i. The fractal dimension index is based on the patch perimeter and the patch area and describes the patch complexity.

**Units:** None

**Range:** FRAC_SD>= 0

**Behaviour:** Equals FRAC_SD = 0 if the fractal dimension index is identical for all patches. Increases, without limit, as the variation of the fractal dimension indices increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Mandelbrot, B. B. 1977. Fractals: Form, Chance, and Dimension. San Francisco. W. H. Freeman and Company.

## See Also

`lsm_p_frac`, `sd`,
`lsm_c_frac_mn`, `lsm_c_frac_cv`,
`lsm_l_frac_mn`, `lsm_l_frac_sd`, `lsm_l_frac_cv`

## Examples

```
lsm_c_frac_sd(landscape)
```

## lsm_c_gyrate_cv *GYRATE_CV (class level)*

### Description

Coefficient of variation radius of gyration (Area and edge metric)

### Usage

```
lsm_c_gyrate_cv(landscape, directions, cell_center)

## S3 method for class 'RasterLayer'
lsm_c_gyrate_cv(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'RasterStack'
lsm_c_gyrate_cv(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'RasterBrick'
lsm_c_gyrate_cv(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'stars'
lsm_c_gyrate_cv(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'list'
lsm_c_gyrate_cv(landscape, directions = 8, cell_center = FALSE)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `cell_center` | If true, the coordinates of the centroid are forced to be a cell center within the patch. |

### Details

$$GYRATE_{CV} = cv(GYRATE[patch_{ij}])$$

where $GYRATE[patch_{ij}]$ equals the radius of gyration of each patch.

GYRATE_CV is an 'Area and edge metric'. The metric summarises each class as the Coefficient of variation of the radius of gyration of all patches belonging to class i. GYRATE measures the distance from each cell to the patch centroid and is based on cell center-to-cell center distances. The metrics characterises both the patch area and compactness. The Coefficient of variation is scaled to the mean and comparable among different landscapes.

If `cell_center = TRUE` some patches might have several possible cell-center centroids. In this case, the gyrate index is based on the mean distance of all cells to all possible cell-center centroids.

**Units:** Meters

**Range:** GYRATE_CV >= 0

**Behaviour:** Equals GYRATE_CV = 0 if the radius of gyration is identical for all patches. Increases, without limit, as the variation of the radius of gyration increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Keitt, T. H., Urban, D. L., & Milne, B. T. 1997. Detecting critical scales in fragmented landscapes. Conservation ecology, 1(1).

## See Also

`lsm_p_gyrate`, `cv`,
`lsm_c_gyrate_mn`, `lsm_c_gyrate_sd`,
`lsm_l_gyrate_mn`, `lsm_l_gyrate_sd`, `lsm_l_gyrate_cv`

## Examples

```
lsm_c_gyrate_cv(landscape)
```

---

# lsm_c_gyrate_mn *GYRATE_MN (class level)*

---

## Description

Mean radius of gyration (Area and edge metric)

## Usage

```
lsm_c_gyrate_mn(landscape, directions, cell_center)

## S3 method for class 'RasterLayer'
lsm_c_gyrate_mn(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'RasterStack'
lsm_c_gyrate_mn(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'RasterBrick'
```

```
lsm_c_gyrate_mn(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'stars'
lsm_c_gyrate_mn(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'list'
lsm_c_gyrate_mn(landscape, directions = 8, cell_center = FALSE)
```

### Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| cell_center | If true, the coordinates of the centroid are forced to be a cell center within the patch. |

### Details

$$GYRATE_{MN} = mean(GYRATE[patch_{ij}])$$

where $GYRATE[patch_{ij}]$ equals the radius of gyration of each patch.

GYRATE_MN is an 'Area and edge metric'. The metric summarises each class as the mean of the radius of gyration of all patches belonging to class i. GYRATE measures the distance from each cell to the patch centroid and is based on cell center-to-cell center distances. The metrics characterises both the patch area and compactness.

If cell_center = TRUE some patches might have several possible cell-center centroids. In this case, the gyrate index is based on the mean distance of all cells to all possible cell-center centroids.

**Units:** Meters

**Range:** GYRATE_MN >= 0

**Behaviour:** Approaches GYRATE_MN = 0 if every patch is a single cell. Increases, without limit, when only one patch is present.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Keitt, T. H., Urban, D. L., & Milne, B. T. 1997. Detecting critical scales in fragmented landscapes. Conservation ecology, 1(1).

## See Also

lsm_p_gyrate, mean,
lsm_c_gyrate_sd, lsm_c_gyrate_cv,
lsm_l_gyrate_mn, lsm_l_gyrate_sd, lsm_l_gyrate_cv

## Examples

```
lsm_c_gyrate_mn(landscape)
```

# lsm_c_gyrate_sd *GYRATE_SD (class level)*

## Description

Standard deviation radius of gyration (Area and edge metric)

## Usage

```
lsm_c_gyrate_sd(landscape, directions, cell_center)

## S3 method for class 'RasterLayer'
lsm_c_gyrate_sd(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'RasterStack'
lsm_c_gyrate_sd(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'RasterBrick'
lsm_c_gyrate_sd(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'stars'
lsm_c_gyrate_sd(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'list'
lsm_c_gyrate_sd(landscape, directions = 8, cell_center = FALSE)
```

## Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| cell_center | If true, the coordinates of the centroid are forced to be a cell center within the patch. |

## Details

$$GYRATE_{SD} = sd(GYRATE[patch_{ij}])$$

where $GYRATE[patch_{ij}]$ equals the radius of gyration of each patch.

GYRATE_SD is an 'Area and edge metric'. The metric summarises each class as the standard deviation of the radius of gyration of all patches belonging to class i. GYRATE measures the distance from each cell to the patch centroid and is based on cell center-to-cell center distances. The metrics characterises both the patch area and compactness.

If `cell_center` = `TRUE` some patches might have several possible cell-center centroids. In this case, the gyrate index is based on the mean distance of all cells to all possible cell-center centroids.

**Units:** Meters

**Range:** GYRATE_SD >= 0

**Behaviour:** Equals GYRATE_SD = 0 if the radius of gyration is identical for all patches. Increases, without limit, as the variation of the radius of gyration increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Keitt, T. H., Urban, D. L., & Milne, B. T. 1997. Detecting critical scales in fragmented landscapes. Conservation ecology, 1(1).

## See Also

`lsm_p_gyrate`, `cv`,
`lsm_c_gyrate_mn`, `lsm_c_gyrate_cv`,
`lsm_l_gyrate_mn`, `lsm_l_gyrate_sd`, `lsm_l_gyrate_cv`

## Examples

```
lsm_c_gyrate_sd(landscape)
```

## lsm_c_iji *Interspersion and Juxtaposition index (class level)*

### Description

Interspersion and Juxtaposition index (Aggregation metric)

### Usage

```
lsm_c_iji(landscape, verbose)

## S3 method for class 'RasterLayer'
lsm_c_iji(landscape, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_c_iji(landscape, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_c_iji(landscape, verbose = TRUE)

## S3 method for class 'stars'
lsm_c_iji(landscape, verbose = TRUE)

## S3 method for class 'list'
lsm_c_iji(landscape, verbose = TRUE)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `verbose` | Print warning message if not sufficient patches are present |

### Details

$$IJI = \frac{- \sum\limits_{k=1}^{m} \left[ \left( \frac{e_{ik}}{\sum\limits_{k=1}^{m} e_{ik}} \right) ln \left( \frac{e_{ik}}{\sum\limits_{k=1}^{m} e_{ik}} \right) \right]}{ln(m - 1)} * 100$$

where $e_{ik}$ are the unique adjacencies of all classes (lower/upper triangle of the adjacency table - without the diagonal) and $m$ is the number of classes.

IJI is an 'Aggregation metric'. It is a so called "salt and pepper" metric and describes the intermixing of classes (i.e. without considering like adjacencies - the diagonal of the adjacency table). The number of classes to calculate IJI must be >= than 3.

**Units:** Percent

**Range:** 0 < IJI <= 100

**Behaviour:** Approaches 0 if a class is only adjacent to a single other class and equals 100 when a class is equally adjacent to all other classes.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

McGarigal, K., & Marks, B. J. 1995. FRAGSTATS: spatial pattern analysis program for quantifying landscape structure. Gen. Tech. Rep. PNW-GTR-351. Portland, OR: US Department of Agriculture, Forest Service, Pacific Northwest Research Station. 122 p, 351.

## See Also

lsm_l_iji

## Examples

```
lsm_c_iji(landscape)
```

---

# lsm_c_lpi *LPI (class level)*

---

## Description

Largest patch index (Area and Edge metric)

## Usage

```
lsm_c_lpi(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_lpi(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_lpi(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_lpi(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_lpi(landscape, directions = 8)
```

```
## S3 method for class 'list'
lsm_c_lpi(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$LPI = \frac{\max_{j=1}^{n}(a_{ij})}{A} * 100$$

where $max(a_{ij})$ is the area of the patch in square meters and $A$ is the total landscape area in square meters.

The largest patch index is an 'Area and edge metric'. It is the percentage of the landscape covered by the corresponding largest patch of each class i. It is a simple measure of dominance.

**Units:** Percentage

**Range:** 0 < LPI <= 100

**Behaviour:** Approaches LPI = 0 when the largest patch is becoming small and equals LPI = 100 when only one patch is present

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_area`, `lsm_l_ta`,
`lsm_l_lpi`

## Examples

```
lsm_c_lpi(landscape)
```

## lsm_c_lsi *LSI (class level)*

### Description

Landscape shape index (Aggregation metric)

### Usage

```
lsm_c_lsi(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_lsi(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_lsi(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_lsi(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_lsi(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_lsi(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$LSI = \frac{e_i}{\min e_i}$$

where $e_i$ is the total edge length in cell surfaces and $\min e_i$ is the minimum total edge length in cell surfaces

LSI is an 'Aggregation metric'. It is the ratio between the actual edge length of class i and the hypothetical minimum edge length of class i. The minimum edge length equals the edge length if class i would be maximally aggregated.

**Units:** None

**Ranges:** LSI >= 1

**Behaviour:** Equals LSI = 1 when only one squared patch is present or all patches are maximally aggregated. Increases, without limit, as the length of the actual edges increases, i.e. the patches become less compact.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

## See Also

`lsm_p_shape`,
`lsm_l_lsi`

## Examples

```
lsm_c_lsi(landscape)
```

---

# lsm_c_mesh *MESH (class level)*

---

## Description

Effective Mesh Size (Aggregation metric)

## Usage

```
lsm_c_mesh(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_mesh(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_mesh(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_mesh(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_mesh(landscape, directions = 8)
```

```
## S3 method for class 'list'
lsm_c_mesh(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$MESH = \frac{\sum_{j=1}^{n} a_{ij}^{2}}{A} * \frac{1}{10000}$$

where $a_{ij}$ is the patch area in square meters and $A$ is the total landscape area in square meters.

The effective mesh size is an 'Aggregation metric'. Because each patch is squared before the sums for each group i are calculated and the sum is standardized by the total landscape area, MESH is a relative measure of patch structure. MESH is perfectly, negatively correlated to `lsm_c_division`.

**Units:** Hectares

**Range:** cell size / total area <= MESH <= total area

**Behaviour:** Equals cellsize/total area if class covers only one cell and equals total area if only one patch is present.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Jaeger, J. A. 2000. Landscape division, splitting index, and effective mesh size: new measures of landscape fragmentation. Landscape ecology, 15(2), 115-130.

## See Also

`lsm_p_area`, `lsm_l_ta`,
`lsm_l_mesh`

## Examples

```
lsm_c_mesh(landscape)
```

## lsm_c_ndca *NDCA (class level)*

### Description

Number of disjunct core areas (Core area metric)

### Usage

```
lsm_c_ndca(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_ndca(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_c_ndca(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_c_ndca(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_c_ndca(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_c_ndca(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `consider_boundary` | Logical if cells that only neighbour the landscape boundary should be considered as core |
| `edge_depth` | Distance (in cells) a cell has the be away from the patch edge to be considered as core cell |

## Details

$$NDCA = \sum_{j=1}^{n} n_{ij}^{core}$$

where $n_{ij}^{core}$ is the number of disjunct core areas.

NDCA is a 'Core area metric'. The metric summarises class i as the sum of all patches belonging to class i. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). NDCA counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells. It describes patch area and shape simultaneously (more core area when the patch is large, however, the shape must allow disjunct core areas). Thereby, a compact shape (e.g. a square) will contain less disjunct core areas than a more irregular patch.

**Units:** None

**Range:** NDCA >= 0

**Behaviour:** NDCA = 0 when TCA = 0, i.e. every cell in patches of class i is an edge. NDCA increases, with out limit, as core area increases and patch shapes allow disjunct core areas (i.e. patch shapes become rather complex).

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

### See Also

lsm_c_tca,
lsm_p_ncore, lsm_l_ndca

### Examples

```
lsm_c_ndca(landscape)
```

---

## lsm_c_nlsi *nLSI (class level)*

---

### Description

Normalized landscape shape index (Aggregation metric)

### Usage

```
lsm_c_nlsi(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_nlsi(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_nlsi(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_nlsi(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_nlsi(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_nlsi(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$nLSI = \frac{e_i}{\min e_i}$$

where $e_i$ is the total edge length in cell surfaces and $\min e_i$ is the minimum total edge length in cell surfaces nLSI is an 'Aggregation metric'. It is the ratio between the actual edge length of class i and the hypothetical minimum edge length of class i. The minimum edge length equals the edge length if class i would be maximally aggregated.

**Units:** None

**Ranges:** nlsi >= 1

**Behaviour:** Equals nlsi = 1 when only one squared patch is present or all patches are maximally aggregated. Increases, without limit, as the length of the actual edges increases, i.e. the patches become less compact.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

### See Also

lsm_p_shape

### Examples

```
lsm_c_nlsi(landscape)
```

---

## lsm_c_np *NP (class level)*

---

### Description

Number of patches (Aggregation metric)

### Usage

```
lsm_c_np(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_np(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_np(landscape, directions = 8)
```

```
## S3 method for class 'RasterBrick'
lsm_c_np(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_np(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_np(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$NP = n_i$$

where $n_i$ is the number of patches.

NP is an 'Aggregation metric'. It describes the fragmentation of a class, however, does not necessarily contain information about the configuration or composition of the class.

**Units:** None

**Ranges:** NP >= 1

**Behaviour:** Equals NP = 1 when only one patch is present and increases, without limit, as the number of patches increases

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

### See Also

`lsm_l_np`

### Examples

```
lsm_c_np(landscape)
```

## lsm_c_pafrac *PAFRAC (class level)*

### Description

Perimeter-Area Fractal Dimension (Shape metric)

### Usage

```
lsm_c_pafrac(landscape, directions, verbose)

## S3 method for class 'RasterLayer'
lsm_c_pafrac(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_c_pafrac(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_c_pafrac(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'stars'
lsm_c_pafrac(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_c_pafrac(landscape, directions = 8, verbose = TRUE)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `verbose` | Print warning message if not sufficient patches are present |

### Details

$$PAFRAC = \frac{2}{\beta}$$

where $\beta$ is the slope of the regression of the area against the perimeter (logarithm) $n_i \sum\limits_{j=1}^{n} \ln a_{ij} = a + \beta n_i \sum\limits_{j=1}^{n} \ln p_{ij}$

PAFRAC is a 'Shape metric'. It describes the patch complexity of class i while being scale independent. This means that increasing the patch size while not changing the patch form will not change the metric. However, it is only meaningful if the relationship between the area and perimeter is linear on a logarithmic scale. Furthermore, if there are less than 10 patches in class i, the metric returns NA because of the small-sample issue.

**Units:** None

**Range:** 1 <= PAFRAC <= 2

**Behaviour:** Approaches PAFRAC = 1 for patches with simple shapes and approaches PAFRAC = 2 for irregular shapes

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Burrough, P. A. 1986. Principles of Geographical Information Systems for Land Resources Assessment. Monographs on Soil and Resources Survey No. 12. Clarendon Press, Oxford

## See Also

lsm_p_area, lsm_p_perim,
lsm_l_pafrac

## Examples

```
lsm_c_pafrac(landscape)
```

---

# lsm_c_para_cv *PARA_CV (class level)*

---

## Description

Coefficient of variation perimeter-area ratio (Shape metric)

## Usage

```
lsm_c_para_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_para_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_para_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_para_cv(landscape, directions = 8)
```

```
## S3 method for class 'stars'
lsm_c_para_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_para_cv(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$PARA_{CV} = cv(PARA[patch_{ij}]$$

where $PARA[patch_{ij}]$ is the perimeter area ratio of each patch.

PARA_CV is a 'Shape metric'. It summarises each class as the Coefficient of variation of each patch belonging to class i. The perimeter-area ratio describes the patch complexity in a straightforward way. However, because it is not standarised to a certain shape (e.g. a square), it is not scale independent, meaning that increasing the patch size while not changing the patch form will change the ratio.

**Units:** None

**Range:** PARA_CV >= 0

**Behaviour:** Equals PARA_CV = 0 if the perimeter-area ratio is identical for all patches. Increases, without limit, as the variation of the perimeter-area ratio increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_para`, `cv`,
`lsm_c_para_mn`, `lsm_c_para_sd`,
`lsm_l_para_mn`, `lsm_l_para_sd`, `lsm_l_para_cv`

## Examples

```
lsm_c_para_cv(landscape)
```

## lsm_c_para_mn *PARA_MN (class level)*

### Description

Mean perimeter-area ratio (Shape metric)

### Usage

```
lsm_c_para_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_para_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_para_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_para_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_para_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_para_mn(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$PARA_{MN} = mean(PARA[patch_{ij}]$$

where $PARA[patch_{ij}]$ is the perimeter area ratio of each patch.

PARA_MN is a 'Shape metric'. It summarises each class as the mean of each patch belonging to class i. The perimeter-area ratio describes the patch complexity in a straightforward way. However, because it is not standarised to a certain shape (e.g. a square), it is not scale independent, meaning that increasing the patch size while not changing the patch form will change the ratio.

**Units:** None

**Range:** PARA_MN > 0

**Behaviour:** Approaches PARA_MN > 0 if PARA for each patch approaches PARA > 0, i.e. the form approaches a rather small square. Increases, without limit, as PARA increases, i.e. patches become more complex.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_para, mean,
lsm_c_para_sd, lsm_c_para_cv,
lsm_l_para_mn, lsm_l_para_sd, lsm_l_para_cv

## Examples

```
lsm_c_para_mn(landscape)
```

---

# lsm_c_para_sd *PARA_SD (class level)*

---

## Description

Standard deviation perimeter-area ratio (Shape metric)

## Usage

```
lsm_c_para_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_para_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_para_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_para_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_para_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_para_sd(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$PARA_{SD} = sd(PARA[patch_{ij}]$$

where $PARA[patch_{ij}]$ is the perimeter area ratio of each patch.

PARA_SD is a 'Shape metric'. It summarises each class as the standard deviation of each patch belonging to class i. The perimeter-area ratio describes the patch complexity in a straightforward way. However, because it is not standarised to a certain shape (e.g. a square), it is not scale independent, meaning that increasing the patch size while not changing the patch form will change the ratio.

**Units:** None

**Range:** PARA_SD >= 0

**Behaviour:** Equals PARA_SD = 0 if the perimeter-area ratio is identical for all patches. Increases, without limit, as the variation of the perimeter-area ratio increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_para`, `sd`,
`lsm_c_para_mn`, `lsm_c_para_cv`,
`lsm_l_para_mn`, `lsm_l_para_sd`, `lsm_l_para_cv`

## Examples

```
lsm_c_para_sd(landscape)
```

## lsm_c_pd *PD (class level)*

### Description

Patch density (Aggregation metric)

### Usage

```
lsm_c_pd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_pd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_pd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_pd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_pd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_pd(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$PD = \frac{n_i}{A} * 10000 * 100$$

where $n_i$ is the number of patches and $A$ is the total landscape area in square meters.

PD is an 'Aggregation metric'. It describes the fragmentation of a class, however, does not necessarily contain information about the configuration or composition of the class. In contrast to `lsm_c_np` it is standardized to the area and comparisons among landscapes with different total area are possible.

**Units:** Number per 100 hectares

**Ranges:** 0 < PD <= 1e+06

**Behaviour:** Increases as the landscape gets more patchy. Reaches its maximum if every cell is a different patch.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

### See Also

lsm_c_np, lsm_l_ta,
lsm_l_pd

### Examples

```
lsm_c_pd(landscape)
```

## lsm_c_pladj *PLADJ (class level)*

### Description

Percentage of Like Adjacencies (Aggregation metric)

### Usage

```
lsm_c_pladj(landscape)

## S3 method for class 'RasterLayer'
lsm_c_pladj(landscape)

## S3 method for class 'RasterStack'
lsm_c_pladj(landscape)

## S3 method for class 'RasterBrick'
lsm_c_pladj(landscape)

## S3 method for class 'stars'
lsm_c_pladj(landscape)

## S3 method for class 'list'
lsm_c_pladj(landscape)
```

### Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |

## Details

$$PLADJ = (\frac{g_{ij}}{\sum\limits_{k=1}^{m} g_{ik}}) * 100$$

where $g_{ii}$ is the number of adjacencies between cells of class i and $g_{ik}$ is the number of adjacencies between cells of class i and k.

PLADJ is an 'Aggregation metric'. It calculates the frequency how often patches of different classes i (focal class) and k are next to each other, and following is a measure of class aggregation. The adjacencies are counted using the double-count method.

**Units:** Percent

**Ranges:** 0 <= PLADJ <= 100

**Behaviour:** Equals PLADJ = 0 if class i is maximal disaggregated, i.e. every cell is a different patch. Equals PLADJ = 100 when the only one patch is present.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## Examples

```
lsm_c_pladj(landscape)
```

---

# `lsm_c_pland` *PLAND (class level)*

---

## Description

Percentage of landscape of class (Area and Edge metric)

## Usage

```
lsm_c_pland(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_pland(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_pland(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_pland(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_pland(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_pland(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$PLAND = \frac{\sum\limits_{j=1}^{n} a_{ij}}{A} * 100$$

where $a_{ij}$ is the area of each patch and $A$ is the total landscape area.

PLAND is an 'Area and edge metric'. It is the percentage of the landscape belonging to class i. It is a measure of composition and because of the relative character directly comparable among landscapes with different total areas.

**Units:** Percentage

**Range:** 0 < PLAND <= 100

**Behaviour:** Approaches PLAND = 0 when the proportional class area is decreasing. Equals PLAND = 100 when only one patch is present.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_c_ca, lsm_l_ta

## Examples

```
lsm_c_pland(landscape)
```

---

# lsm_c_shape_cv *SHAPE_CV (class level)*

---

## Description

Covariance of variation shape index (Shape metric)

## Usage

```
lsm_c_shape_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_shape_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_shape_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_shape_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_shape_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_shape_cv(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$SHAPE_{CV} = cv(SHAPE[patch_{ij}])$$

where $SHAPE[patch_{ij}]$ is the shape index of each patch.

SHAPE_CV is a 'Shape metric'. Each class is summarised as the Coefficient of variation of each patch belonging to class i. SHAPE describes the ratio between the actual perimeter of the patch and the hypothetical minimum perimeter of the patch. The minimum perimeter equals the perimeter if the patch would be maximally compact.

**Units:** None

**Range:** SHAPE_CV >= 0

**Behaviour:** Equals SHAPE_CV = 0 if all patches have an identical shape index. Increases, without limit, as the variation of the shape index increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

## See Also

`lsm_p_shape`, `cv`,
`lsm_c_shape_mn`, `lsm_c_shape_sd`,
`lsm_l_shape_mn`, `lsm_l_shape_sd`, `lsm_l_shape_cv`

## Examples

```
lsm_c_shape_cv(landscape)
```

## lsm_c_shape_mn *SHAPE_MN (class level)*

### Description

Mean shape index (Shape metric)

### Usage

```
lsm_c_shape_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_shape_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_shape_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_shape_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_shape_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_shape_mn(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$SHAPE_{MN} = mean(SHAPE[patch_{ij}])$$

where $SHAPE[patch_{ij}]$ is the shape index of each patch.

SHAPE_MN is a 'Shape metric'. Each class is summarised as the mean of each patch belonging to class i. SHAPE describes the ratio between the actual perimeter of the patch and the hypothetical minimum perimeter of the patch. The minimum perimeter equals the perimeter if the patch would be maximally compact.

**Units:** None

**Range:** SHAPE_SD >= 1

**Behaviour:** Equals SHAPE_MN = 0 if all patches are squares. Increases, without limit, as the shapes of patches become more complex.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

## See Also

`lsm_p_shape`, `mean`,
`lsm_c_shape_sd`, `lsm_c_shape_cv`,
`lsm_l_shape_mn`, `lsm_l_shape_sd`, `lsm_l_shape_cv`

## Examples

```
lsm_c_shape_mn(landscape)
```

---

# lsm_c_shape_sd *SHAPE_SD (class level)*

---

## Description

Standard deviation shape index (Shape metric)

## Usage

```
lsm_c_shape_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_shape_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_shape_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_shape_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_shape_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_shape_sd(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$SHAPE_{SD} = sd(SHAPE[patch_{ij}])$$

where $SHAPE[patch_{ij}]$ is the shape index of each patch.

SHAPE_SD is a 'Shape metric'. Each class is summarised as the standard deviation of each patch belonging to class i. SHAPE describes the ratio between the actual perimeter of the patch and the hypothetical minimum perimeter of the patch. The minimum perimeter equals the perimeter if the patch would be maximally compact.

**Units:** None

**Range:** SHAPE_SD >= 0

**Behaviour:** Equals SHAPE_SD = 0 if all patches have an identical shape index. Increases, without limit, as the variation of the shape index increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

## See Also

`lsm_p_shape`, `sd`,
`lsm_c_shape_mn`, `lsm_c_shape_cv`,
`lsm_l_shape_mn`, `lsm_l_shape_sd`, `lsm_l_shape_cv`

## Examples

```
lsm_c_shape_sd(landscape)
```

## lsm_c_split *SPLIT (class level)*

### Description

Splitting index (Aggregation metric)

### Usage

```
lsm_c_split(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_c_split(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_split(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_split(landscape, directions = 8)

## S3 method for class 'stars'
lsm_c_split(landscape, directions = 8)

## S3 method for class 'list'
lsm_c_split(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$SPLIT = \frac{A^2}{\sum\limits_{j=1}^{n} a_{ij}^2}$$

where $a_{ij}$ is the patch area in square meters and $A$ is the total landscape area.

SPLIT is an 'Aggregation metric'. It describes the number of patches if all patches of class i would be divided into equally sized patches.

**Units:** None

**Range:** 1 <= SPLIT <= Number of cells squared

**Behaviour:** Equals SPLIT = 1 if only one patch is present. Increases as the number of patches of class i increases and is limited if all cells are a patch

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Jaeger, J. A. 2000. Landscape division, splitting index, and effective mesh size: new measures of landscape fragmentation. Landscape ecology, 15(2), 115-130.

### See Also

lsm_p_area, lsm_l_ta,
lsm_l_split

### Examples

```
lsm_c_split(landscape)
```

## lsm_c_tca *TCA (class level)*

### Description

Total core area (Core area metric)

### Usage

```
lsm_c_tca(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_c_tca(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_c_tca(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_c_tca(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_c_tca(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_c_tca(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$TCA = \sum_{j=1}^{n} a_{ij}^{core} * (\frac{1}{10000})$$

where here $a_{ij}^{core}$ is the core area in square meters.

TCA is a 'Core area metric' and equals the sum of core areas of all patches belonging to class i. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). In other words, the core area of a patch is all area that is not an edge. It characterises patch areas and shapes of patches belonging to class i simultaneously (more core area when the patch is large and the shape is rather compact, i.e. a square). Additionally, TCA is a measure for the configuration of the landscape, because the sum of edges increase as patches are less aggregated.

**Units:** Hectares

**Range:** TCA >= 0

**Behaviour:** Increases, without limit, as patch areas increase and patch shapes simplify. TCA = 0 when every cell in every patch of class i is an edge.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_core`, `lsm_l_tca`

## Examples

```
lsm_c_tca(landscape)
```

## lsm_c_te *TE (class level)*

### Description

Total (class) edge (Area and Edge metric)

### Usage

```
lsm_c_te(landscape, count_boundary, directions)

## S3 method for class 'RasterLayer'
lsm_c_te(landscape, count_boundary = FALSE, directions = 8)

## S3 method for class 'RasterStack'
lsm_c_te(landscape, count_boundary = FALSE, directions = 8)

## S3 method for class 'RasterBrick'
lsm_c_te(landscape, count_boundary = FALSE, directions = 8)

## S3 method for class 'stars'
lsm_c_te(landscape, count_boundary = FALSE, directions = 8)

## S3 method for class 'list'
lsm_c_te(landscape, count_boundary = FALSE, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `count_boundary` | Include landscape boundary in edge length |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$TE = \sum_{k=1}^{m} e_{ik}$$

where $e_{ik}$ is the edge lengths in meters. TE is an 'Area and edge metric'. Total (class) edge includes all edges between class i and all other classes k. It measures the configuration of the landscape because a highly fragmented landscape will have many edges. However, total edge is an absolute measure, making comparisons among landscapes with different total areas difficult. If `cound_boundary = TRUE` also edges to the landscape boundary are included.

**Units:** Meters

**Range:** TE >= 0

**Behaviour:** Equals TE = 0 if all cells are edge cells. Increases, without limit, as landscape becomes more fragmented

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_perim lsm_l_te

## Examples

```
lsm_c_te(landscape)
```

---

# lsm_l_ai *AI (landscape level)*

---

## Description

Aggregation index (Aggregation metric)

## Usage

```
lsm_l_ai(landscape)

## S3 method for class 'RasterLayer'
lsm_l_ai(landscape)

## S3 method for class 'RasterStack'
lsm_l_ai(landscape)

## S3 method for class 'RasterBrick'
lsm_l_ai(landscape)

## S3 method for class 'stars'
lsm_l_ai(landscape)

## S3 method for class 'list'
lsm_l_ai(landscape)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers |

## Details

$$AI = \left[ \sum_{i=1}^{m} \Big( \frac{g_{ii}}{max - g_{ii}} \Big) P_i \right] (100)$$

where $g_{ii}$ is the number of like adjacencies based on the single-count method and $max - g_{ii}$ is the classwise maximum number of like adjacencies of class i and $P_i$ the proportion of landscape compromised of class i.

AI is an 'Aggregation metric'. It equals the number of like adjacencies divided by the theoretical maximum possible number of like adjacencies for that class summed over each class for the entire landscape. The metric is based on he adjacency matrix and the the single-count method.

**Units:** Percent

**Range:** 0 <= AI <= 100

**Behaviour:** Equals 0 for maximally disaggregated and 100 for maximally aggregated classes.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

He, H. S., DeZonia, B. E., & Mladenoff, D. J. 2000. An aggregation index (AI) to quantify spatial patterns of landscapes. Landscape ecology, 15(7), 591-601.

## See Also

`lsm_c_ai`

## Examples

```
lsm_l_ai(landscape)
```

## lsm_l_area_cv *AREA_CV (landscape level)*

### Description

Coefficient of variation of patch area (Area and edge metric)

### Usage

```
lsm_l_area_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_area_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_area_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_area_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_area_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_area_cv(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$AREA_{CV} = cv(AREA[patch_{ij}])$$

where $AREA[patch_{ij}]$ is the area of each patch in hectares.

AREA_CV is an 'Area and Edge metric'. The metric summarises the landscape as the Coefficient of variation of all patches in the landscape. The metric describes the differences among patches in the landscape and is easily comparable because it is scaled to the mean.

**Units:** Hectares

**Range:** AREA_CV >= 0

**Behaviour:** Equals AREA_CV = 0 if all patches are identical in size. Increases, without limit, as the variation of patch areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_area, cv,
lsm_c_area_mn, lsm_c_area_sd, lsm_c_area_cv,
lsm_l_area_mn, lsm_l_area_sd

## Examples

```
lsm_l_area_cv(landscape)
```

---

# lsm_l_area_mn *AREA_MN (landscape level)*

---

## Description

Mean of patch area (Area and edge metric)

## Usage

```
lsm_l_area_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_area_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_area_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_area_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_area_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_area_mn(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$AREA_{MN} = mean(AREA[patch_{ij}])$$

where $AREA[patch_{ij}]$ is the area of each patch in hectares

AREA_MN is an 'Area and Edge metric'. The metric summarises the landscape as the mean of all patch in the landscape. The metric is a simple way to describe the composition of the landscape. Especially together with the total landscape area (`lsm_l_ta`), it can also give an an idea of patch structure (e.g. many small patches vs. few larges patches).

**Units:** Hectares

**Range:** AREA_MN > 0

**Behaviour:** Approaches AREA_MN = 0 if all patches are small. Increases, without limit, as the patch areas increase.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_area`, `mean`,
`lsm_c_area_mn`, `lsm_c_area_sd`, `lsm_c_area_cv`
`lsm_l_area_sd`, `lsm_l_area_cv`

## Examples

```
lsm_l_area_mn(landscape)
```

## lsm_l_area_sd AREA_SD (landscape level)

### Description

Standard deviation of patch area (Area and edge metric)

### Usage

```
lsm_l_area_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_area_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_area_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_area_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_area_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_area_sd(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$AREA_{SD} = sd(AREA[patch_{ij}])$$

where $AREA[patch_{ij}]$ is the area of each patch in hectares.

AREA_SD is an 'Area and Edge metric'. The metric summarises the landscape as the standard deviation of all patch in the landscape. The metric describes the differences among all patches in the landscape.

**Units:** Hectares

**Range:** AREA_SD >= 0

**Behaviour:** Equals AREA_SD = 0 if all patches are identical in size. Increases, without limit, as the variation of patch areas increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

### See Also

lsm_p_area, sd,
lsm_c_area_mn, lsm_c_area_sd, lsm_c_area_cv
lsm_l_area_mn, lsm_l_area_cv

### Examples

```
lsm_l_area_sd(landscape)
```

---

## lsm_l_cai_cv *CAI_CV (landscape level)*

---

### Description

Coefficient of variation of core area index (Core area metric)

### Usage

```
lsm_l_cai_cv(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_cai_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_l_cai_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

```
## S3 method for class 'RasterBrick'
lsm_l_cai_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_l_cai_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_l_cai_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `consider_boundary` | Logical if cells that only neighbour the landscape boundary should be considered as core |
| `edge_depth` | Distance (in cells) a cell has the be away from the patch edge to be considered as core cell |

## Details

$$CAI_{CV} = cv(CAI[patch_{ij}]$$

where $CAI[patch_{ij}]$ is the core area index of each patch.

CAI_CV is a 'Core area metric'. The metric summarises the landscape as the Coefficient of variation of the core area index of all patches in the landscape. The core area index is the percentage of core area in relation to patch area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). The metric describes the differences among all patches in the landscape. Because it is scaled to the mean, it is easily comparable.

**Units:** Percent

**Range:** CAI_CV >= 0

**Behaviour:** Equals CAI_CV = 0 if the core area index is identical for all patches. Increases, without limit, as the variation of the core area indices increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_cai, cv,
lsm_c_cai_mn, lsm_c_cai_sd, lsm_c_cai_cv,
lsm_l_cai_mn, lsm_l_cai_sd

## Examples

```
lsm_l_cai_cv(landscape)
```

---

# lsm_l_cai_mn *CAI_MN (landscape level)*

---

## Description

Mean of core area index (Core area metric)

## Usage

```
lsm_l_cai_mn(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_cai_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_l_cai_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_l_cai_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_l_cai_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_l_cai_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| consider_boundary | Logical if cells that only neighbour the landscape boundary should be considered as core |
| edge_depth | Distance (in cells) a cell has the be away from the patch edge to be considered as core cell |

## Details

$$CAI_{MN} = mean(CAI[patch_{ij}]$$

where $CAI[patch_{ij}]$ is the core area index of each patch.

CAI_MN is a 'Core area metric'. The metric summarises the landscape as the mean of the core area index of all patches in the landscape. The core area index is the percentage of core area in relation to patch area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case).

**Units:** Percent

**Range:** 0 <= CAI_MN <= 100

**Behaviour:** CAI_MN = 0 when all patches have no core area and approaches CAI_MN = 100 with increasing percentage of core area within patches.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_cai, mean,
lsm_c_cai_sd, lsm_c_cai_sd, lsm_c_cai_cv,
lsm_l_cai_sd, lsm_l_cai_cv

## Examples

```
lsm_l_cai_mn(landscape)
```

---

# lsm_l_cai_sd *CAI_SD (landscape level)*

---

## Description

Standard deviation of core area index (Core area metric)

## Usage

```
lsm_l_cai_sd(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_cai_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

```
## S3 method for class 'RasterStack'
lsm_l_cai_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_l_cai_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_l_cai_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_l_cai_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$CAI_{SD} = sd(CAI[patch_{ij}]$$

where $CAI[patch_{ij}]$ is the core area index of each patch.

CAI_SD is a 'Core area metric'. The metric summarises the landscape as the standard deviation of the core area index of all patches in the landscape. The core area index is the percentage of core area in relation to patch area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). The metric describes the differences among all patches in the landscape.

**Units:** Percent

**Range:** CAI_SD >= 0

**Behaviour:** Equals CAI_SD = 0 if the core area index is identical for all patches. Increases, without limit, as the variation of core area indices increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_cai`, `sd`,
`lsm_c_cai_mn`, `lsm_c_cai_sd`, `lsm_c_cai_cv`,
`lsm_l_cai_mn`, `lsm_l_cai_cv`

## Examples

```
lsm_l_cai_sd(landscape)
```

---

# `lsm_l_circle_cv` *CIRCLE_CV (landscape level)*

---

## Description

Coefficient of variation of related circumscribing circle (Shape metric)

## Usage

```
lsm_l_circle_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_circle_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_circle_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_circle_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_circle_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_circle_cv(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$CIRCLE_{CV} = cv(CIRCLE[patch_{ij}])$$

where $CIRCLE[patch_{ij}]$ is the related circumscribing circle of each patch.

CIRCLE_CV is a 'Shape metric' and summarises the landscape as the Coefficient of variation of the related circumscribing circle of all patches in the landscape. CIRCLE describes the ratio between the patch area and the smallest circumscribing circle of the patch and characterises the compactness of the patch. CIRCLE_CV describes the differences among all patches in the landscape. Because it is scaled to the mean, it is easily comparable.

**Units:** None

**Range:** CIRCLE_CV >= 0

**Behaviour:** Equals CIRCLE_CV if the related circumscribing circle is identical for all patches. Increases, without limit, as the variation of related circumscribing circles increases.

## Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Baker, W. L., and Y. Cai. 1992. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. Landscape Ecology 7: 291-302.

Based on C++ code from Project Nayuki (https://www.nayuki.io/page/smallest-enclosing-circle).

### See Also

`lsm_p_circle`, `mean`,
`lsm_c_circle_mn`, `lsm_c_circle_sd`, `lsm_c_circle_cv`,
`lsm_l_circle_mn`, `lsm_l_circle_sd`

### Examples

```
lsm_l_circle_cv(landscape)
```

---

## `lsm_l_circle_mn` *CIRCLE_MN (landscape level)*

---

### Description

Mean of related circumscribing circle (Shape metric)

### Usage

```
lsm_l_circle_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_circle_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_circle_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_circle_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_circle_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_circle_mn(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$CIRCLE_{MN} = mean(CIRCLE[patch_{ij}])$$

where $CIRCLE[patch_{ij}]$ is the related circumscribing circle of each patch.

CIRCLE_MN is a 'Shape metric' and summarises the landscape as the mean of the related circumscribing circle of all patches in the landscape. CIRCLE describes the ratio between the patch area and the smallest circumscribing circle of the patch and characterises the compactness of the patch.

**Units:** None

**Range:** CIRCLE_MN > 0

**Behaviour:** Approaches CIRCLE_MN = 0 if the related circumscribing circle of all patches is small. Increases, without limit, as the related circumscribing circles increase.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Baker, W. L., and Y. Cai. 1992. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. Landscape Ecology 7: 291-302.

Based on C++ code from Project Nayuki (https://www.nayuki.io/page/smallest-enclosing-circle).

## See Also

`lsm_p_circle`, `mean`,
`lsm_c_circle_mn`, `lsm_c_circle_sd`, `lsm_c_circle_cv`,
`lsm_l_circle_sd`, `lsm_l_circle_cv`

## Examples

```
lsm_l_circle_mn(landscape)
```

## lsm_l_circle_sd *CIRCLE_SD (landscape level)*

### Description

Standard deviation of related circumscribing circle (Shape metric)

### Usage

```
lsm_l_circle_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_circle_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_circle_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_circle_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_circle_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_circle_sd(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$CIRCLE_{SD} = sd(CIRCLE[patch_{ij}])$$

where $CIRCLE[patch_{ij}]$ is the related circumscribing circle of each patch.

CIRCLE_SD is a 'Shape metric' and summarises the landscape as the standard deviation of the related circumscribing circle of all patches in the landscape. CIRCLE describes the ratio between the patch area and the smallest circumscribing circle of the patch and characterises the compactness of the patch. The metric describes the differences among all patches of the landscape.

**Units:** None

**Range:** CIRCLE_SD >= 0

**Behaviour:** Equals CIRCLE_SD if the related circumscribing circle is identical for all patches. Increases, without limit, as the variation of related circumscribing circles increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Baker, W. L., and Y. Cai. 1992. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. Landscape Ecology 7: 291-302.

Based on C++ code from Project Nayuki (https://www.nayuki.io/page/smallest-enclosing-circle).

### See Also

`lsm_p_circle`, `mean`,
`lsm_c_circle_mn`, `lsm_c_circle_sd`, `lsm_c_circle_cv`,
`lsm_l_circle_mn`, `lsm_l_circle_cv`

### Examples

```
lsm_l_circle_sd(landscape)
```

---

## lsm_l_cohesion *COHESION (landscape level)*

---

### Description

Patch Cohesion Index (Aggregation metric)

### Usage

```
lsm_l_cohesion(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_cohesion(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_cohesion(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_cohesion(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_cohesion(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_cohesion(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$COHESION = 1 - (\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} \sqrt{a_{ij}}}) * (1 - \frac{1}{\sqrt{Z}})^{-1} * 100$$

where $p_{ij}$ is the perimeter in meters, $a_{ij}$ is the area in square meters and $Z$ is the number of cells.

COHESION is an 'Aggregation metric'.

**Units:** Percent

**Ranges:** Unknown

**Behaviour:** Unknown

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Schumaker, N. H. 1996. Using landscape indices to predict habitat connectivity. Ecology, 77(4), 1210-1225.

## See Also

`lsm_p_perim`, `lsm_p_area`,
`lsm_l_cohesion`

## Examples

```
lsm_l_cohesion(landscape)
```

## lsm_l_condent *Conditional entropy (landscape level)*

### Description

Conditional entropy \[H(y|x)\]

### Usage

```
lsm_l_condent(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")

## S3 method for class 'RasterLayer'
lsm_l_condent(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")

## S3 method for class 'RasterStack'
lsm_l_condent(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")

## S3 method for class 'RasterBrick'
lsm_l_condent(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")

## S3 method for class 'stars'
lsm_l_condent(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")

## S3 method for class 'list'
lsm_l_condent(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `neighbourhood` | The number of directions in which cell adjacencies are considered as neighbours: 4 (rook's case) or 8 (queen's case). The default is 4. |
| `ordered` | The type of pairs considered. Either ordered (TRUE) or unordered (FALSE). The default is TRUE. |
| `base` | The unit in which entropy is measured. The default is "log2", which compute entropy in "bits". "log" and "log10" can be also used. |

### Details

Complexity of a landscape pattern configuration. It measures a only a geometric intricacy (configurational complexity) of a landscape pattern.

### Value

tibble

### References

Nowosad J., TF Stepinski. 2019. Information theory as a consistent framework for quantification and classification of landscape patterns. https://doi.org/10.1007/s10980-019-00830-x

### See Also

lsm_l_ent, lsm_l_mutinf, lsm_l_joinent,

### Examples

```
lsm_l_condent(landscape)
```

---

## lsm_l_contag *CONTAG (landscape level)*

---

### Description

Contagion (Aggregation metric)

### Usage

```
lsm_l_contag(landscape, verbose)

## S3 method for class 'RasterLayer'
lsm_l_contag(landscape, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_l_contag(landscape, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_l_contag(landscape, verbose = TRUE)

## S3 method for class 'stars'
lsm_l_contag(landscape, verbose = TRUE)

## S3 method for class 'list'
lsm_l_contag(landscape, verbose = TRUE)
```

### Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| verbose | Print warning message if not sufficient patches are present |

## Details

$$CONTAG = 1 + \frac{\sum_{q=1}^{n_a} p_q ln(p_q)}{2ln(t)}$$

where $p_q$ the adjacency table for all classes divided by the sum of that table and $t$ the number of classes in the landscape.

CONTAG is an 'Aggregation metric'. It is based on cell adjacencies and describes the probability of two random cells belonging to the same class. $p_q$ is the cell adjacency table, where the order is preserved and pairs of adjacent cells are counted twice. Contagion is affected by both the dispersion and interspersion of classes. E.g., low class dispersion (= high proportion of like adjacencies) and low interspersion (= uneven distribution of pairwise adjacencies) lead to a high contagion value.

The number of classes to calculate CONTAG must be >= than 2.

**Units:** Percent

**Range:** 0 < Contag <=100

**Behaviour:** Approaches CONTAG = 0 if all cells are unevenly distributed and 100 indicates that all cells are equally adjacent to all other classes.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Riitters, K.H., O'Neill, R.V., Wickham, J.D. & Jones, K.B. (1996). A note on contagion indices for landscape analysis. Landscape ecology, 11, 197–202.

## Examples

```
lsm_l_contag(landscape)
```

## lsm_l_contig_cv *CONTIG_CV (landscape level)*

### Description

Coefficient of variation of Contiguity index (Shape metric)

### Usage

```
lsm_l_contig_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_contig_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_contig_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_contig_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_contig_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_contig_cv(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$CONTIG_{CV} = cv(CONTIG[patch_{ij}])$$

where $CONTIG[patch_{ij}]$ is the contiguity of each patch.

CONTIG_CV is a 'Shape metric'. It summarises the landscape as the coefficient of variation of all patches in the landscape. CONTIG_CV asses the spatial connectedness (contiguity) of cells in patches. The metric coerces patch values to a value of 1 and the background to NA. A nine cell focal filter matrix:

```
filter_matrix <- matrix(c(1, 2, 1,
                          2, 1, 2,
                          1, 2, 1), 3, 3, byrow = T)
```

... is then used to weight orthogonally contiguous pixels more heavily than diagonally contiguous pixels. Therefore, larger and more connections between patch cells in the rookie case result in larger contiguity index values.

**Units:** None

**Range:** CONTIG_CV >= 0

**Behaviour:** CONTIG_CV = 0 if the contiguity index is identical for all patches. Increases, without limit, as the variation of CONTIG increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

LaGro, J. 1991. Assessing patch shape in landscape mosaics. Photogrammetric Engineering and Remote Sensing, 57(3), 285-293

## See Also

lsm_p_contig, lsm_c_contig_sd, lsm_c_contig_cv, lsm_c_contig_mn, lsm_l_contig_sd, lsm_l_contig_mn

## Examples

```
lsm_l_contig_cv(landscape)
```

---

# lsm_l_contig_mn *CONTIG_MN (landscape level)*

---

## Description

Mean of Contiguity index (Shape metric)

## Usage

```
lsm_l_contig_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_contig_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
```

```
lsm_l_contig_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_contig_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_contig_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_contig_mn(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$CONTIG_{MN} = mean(CONTIG[patch_i])$$

where $CONTIG[patch_{ij}]$ is the contiguity of each patch.

CONTIG_MN is a 'Shape metric'. It summarises the landscape as the mean of all patches in the landscape. CONTIG_MN asses the spatial connectedness (contiguity) of cells in patches. The metric coerces patch values to a value of 1 and the background to NA. A nine cell focal filter matrix:

```
filter_matrix <- matrix(c(1, 2, 1,
                          2, 1, 2,
                          1, 2, 1), 3, 3, byrow = T)
```

... is then used to weight orthogonally contiguous pixels more heavily than diagonally contiguous pixels. Therefore, larger and more connections between patch cells in the rookie case result in larger contiguity index values.

**Units:** None

**Range:** 0 >= CONTIG_MN <= 1

**Behaviour:** CONTIG equals the mean of the contiguity index on landscape level for all patches.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

LaGro, J. 1991. Assessing patch shape in landscape mosaics. Photogrammetric Engineering and Remote Sensing, 57(3), 285-293

### See Also

lsm_p_contig, lsm_c_contig_sd, lsm_c_contig_cv, lsm_c_contig_mn,
lsm_l_contig_sd, lsm_l_contig_cv

### Examples

```
lsm_l_contig_mn(landscape)
```

## lsm_l_contig_sd CONTIG_SD (landscape level)

### Description

Standard deviation of Contiguity index (Shape metric)

### Usage

```
lsm_l_contig_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_contig_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_contig_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_contig_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_contig_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_contig_sd(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$CONTIG_{SD} = sd(CONTIG[patch_{ij}])$$

where $CONTIG[patch_{ij}]$ is the contiguity of each patch.

CONTIG_SD is a 'Shape metric'. It summarises the landscape as the standard deviation of all patches in the landscape. CONTIG_SD asses the spatial connectedness (contiguity) of cells in patches. The metric coerces patch values to a value of 1 and the background to NA. A nine cell focal filter matrix:

```
filter_matrix <- matrix(c(1, 2, 1,
                          2, 1, 2,
                          1, 2, 1), 3, 3, byrow = T)
```

... is then used to weight orthogonally contiguous pixels more heavily than diagonally contiguous pixels. Therefore, larger and more connections between patch cells in the rookie case result in larger contiguity index values.

**Units:** None

**Range:** CONTIG_SD >= 0

**Behaviour:** CONTIG_SD = 0 if the contiguity index is identical for all patches. Increases, without limit, as the variation of CONTIG increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

LaGro, J. 1991. Assessing patch shape in landscape mosaics. Photogrammetric Engineering and Remote Sensing, 57(3), 285-293

## See Also

lsm_p_contig, lsm_c_contig_sd, lsm_c_contig_cv, lsm_c_contig_mn, lsm_l_contig_cv, lsm_l_contig_mn

## Examples

```
lsm_l_contig_sd(landscape)
```

## lsm_l_core_cv *CORE_CV (landscape level)*

### Description

Coefficient of variation of core area (Core area metric)

### Usage

```
lsm_l_core_cv(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_core_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_l_core_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_l_core_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_l_core_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_l_core_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `consider_boundary` | Logical if cells that only neighbour the landscape boundary should be considered as core |
| `edge_depth` | Distance (in cells) a cell has the be away from the patch edge to be considered as core cell |

## Details

$$CORE_{CV} = cv(CORE[patch_{ij}])$$

where $CORE[patch_{ij}]$ is the core area in square meters of each patch.

CORE_CV is a 'Core area metric'. It equals the Coefficient of variation of the core area of each patch in the landscape. The core area is defined as all cells that have no neighbour with a different value than themselves (rook's case). The metric describes the differences among all patches in the landscape and is easily comparable because it is scaled to the mean.

**Units:** Hectares

**Range:** CORE_CV >= 0

**Behaviour:** Equals CORE_CV = 0 if all patches have the same core area. Increases, without limit, as the variation of patch core areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_core`, `cv`,
`lsm_c_core_mn`, `lsm_c_core_sd`, `lsm_c_core_cv`,
`lsm_l_core_mn`, `lsm_l_core_sd`

## Examples

```
lsm_l_core_cv(landscape)
```

---

`lsm_l_core_mn` *CORE_MN (landscape level)*

---

## Description

Mean of core area (Core area metric)

## Usage

```
lsm_l_core_mn(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_core_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_l_core_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_l_core_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_l_core_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

```
## S3 method for class 'list'
lsm_l_core_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `consider_boundary` | Logical if cells that only neighbour the landscape boundary should be considered as core |
| `edge_depth` | Distance (in cells) a cell has the be away from the patch edge to be considered as core cell |

## Details

$$CORE_{MN} = mean(CORE[patch_{ij}])$$

where $CORE[patch_{ij}]$ is the core area in square meters of each patch.

CORE_MN is a 'Core area metric' and equals the mean of core areas of all patches in the landscape. The core area is defined as all cells that have no neighbour with a different value than themselves (rook's case).

**Units:** Hectares

**Range:** CORE_MN >= 0

**Behaviour:** Equals CORE_MN = 0 if CORE = 0 for all patches. Increases, without limit, as the core area indices increase.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_core`, `mean`,
`lsm_c_core_mn`, `lsm_c_core_sd`, `lsm_c_core_cv`,
`lsm_l_core_sd`, `lsm_l_core_cv`

## Examples

```
lsm_l_core_mn(landscape)
```

---

`lsm_l_core_sd` *CORE_SD (landscape level)*

---

## Description

Standard deviation of patch core area (class level)

## Usage

```
lsm_l_core_sd(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_core_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_l_core_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_l_core_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_l_core_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

```
## S3 method for class 'list'
lsm_l_core_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$CORE_{SD} = sd(CORE[patch_{ij}])$$

where $CORE[patch_{ij}]$ is the core area in square meters of each patch.

CORE_SD is a 'Core area metric'. It equals the standard deviation of the core area of all patches in the landscape. The core area is defined as all cells that have no neighbour with a different value than themselves (rook's case). The metric describes the differences among all patches in the landscape.

**Units:** Hectares

**Range:** CORE_SD >= 0

**Behaviour:** Equals CORE_SD = 0 if all patches have the same core area. Increases, without limit, as the variation of patch core areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_core`, `sd`,
`lsm_c_core_mn`, `lsm_c_core_sd`, `lsm_c_core_cv`,
`lsm_l_core_mn`, `lsm_l_core_cv`

## Examples

```
lsm_l_core_sd(landscape)
```

## lsm_l_dcad *DCAD (landscape level)*

## Description

Disjunct core area density (core area metric)

## Usage

```
lsm_l_dcad(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_dcad(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_l_dcad(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_l_dcad(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_l_dcad(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

```
## S3 method for class 'list'
lsm_l_dcad(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$DCAD = (\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} n_{ij}^{core}}{A}) * 10000 * 100$$

where $n_{ij}^{core}$ is the number of disjunct core areas and $A$ is the total landscape area in square meters.

DCAD is a 'Core area metric'. It equals the number of disjunct core areas per 100 ha relative to the total area. A disjunct core area is a 'patch within the patch' containing only core cells. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). The metric is relative and therefore comparable among landscapes with different total areas.

**Units:** Number per 100 hectares

**Range:** DCAD >= 0

**Behaviour:** Equals DCAD = 0 when DCORE = 0, i.e. no patch contains a disjunct core area. Increases, without limit, as disjunct core areas become more present, i.e. patches becoming larger and less complex.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_c_ndca, lsm_l_ta,
lsm_c_dcad

## Examples

```
lsm_l_dcad(landscape)
```

---

lsm_l_dcore_cv *DCORE_CV (landscape level)*

---

## Description

Coefficient of variation number of disjunct core areas (Core area metric)

## Usage

```
lsm_l_dcore_cv(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_dcore_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_l_dcore_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_l_dcore_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_l_dcore_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_l_dcore_cv(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `consider_boundary` | Logical if cells that only neighbour the landscape boundary should be considered as core |
| `edge_depth` | Distance (in cells) a cell has the be away from the patch edge to be considered as core cell |

## Details

$$DCORE_{CV} = cv(NCORE[patch_{ij}])$$

where $NCORE[patch_{ij}]$ is the number of core areas.

DCORE_CV is an 'Core area metric'. It summarises the landscape as the Coefficient of variation of all patches belonging to the landscape. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). NCORE counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells. The metric describes the differences among all patches in the landscape and is easily comparable because it is scaled to the mean.

**Units:** None

**Range:** DCORE_CV >= 0

**Behaviour:** Equals DCORE_CV = 0 if all patches have the same number of disjunct core areas. Increases, without limit, as the variation of number of disjunct core areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_ncore, cv,
lsm_c_dcore_mn, lsm_c_dcore_sd, lsm_c_dcore_cv,
lsm_l_dcore_mn, lsm_l_dcore_sd

## Examples

```
lsm_l_dcore_cv(landscape)
```

---

# lsm_l_dcore_mn *DCORE_MN (landscape level)*

---

## Description

Mean number of disjunct core areas (Core area metric)

## Usage

```
lsm_l_dcore_mn(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_dcore_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_l_dcore_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_l_dcore_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_l_dcore_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_l_dcore_mn(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$DCORE_{MN} = mean(NCORE[patch_{ij}])$$

where $NCORE[patch_{ij}]$ is the number of core areas.

DCORE_MN is an 'Core area metric'. It summarises the landscape as the mean of all patches in the landscape. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). NCORE counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells.

**Units:** None

**Range:** DCORE_MN > 0

**Behaviour:** Equals DCORE_MN = 0 if NCORE = 0 for all patches. Increases, without limit, as the number of disjunct core areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_ncore, mean,
lsm_c_dcore_mn, lsm_c_dcore_sd, lsm_c_dcore_cv,
lsm_l_dcore_sd, lsm_l_dcore_cv

## Examples

```
lsm_l_dcore_mn(landscape)
```

---

# lsm_l_dcore_sd *DCORE_SD (landscape level)*

---

## Description

Standard deviation number of disjunct core areas (Core area metric)

## Usage

```
lsm_l_dcore_sd(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_dcore_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_l_dcore_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

```
## S3 method for class 'RasterBrick'
lsm_l_dcore_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_l_dcore_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_l_dcore_sd(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `consider_boundary` | Logical if cells that only neighbour the landscape boundary should be considered as core |
| `edge_depth` | Distance (in cells) a cell has the be away from the patch edge to be considered as core cell |

## Details

$$DCORE_{SD} = sd(NCORE[patch_{ij}])$$

where $NCORE[patch_{ij}]$ is the number of core areas.

DCORE_SD is an 'Core area metric'. It summarises the landscape as the standard deviation of all patches. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). NCORE counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells. The metric describes the differences among all patches in the landscape.

**Units:** None

**Range:** DCORE_SD >= 0

**Behaviour:** Equals DCORE_SD = 0 if all patches have the same number of disjunct core areas. Increases, without limit, as the variation of number of disjunct core areas increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_ncore, sd,
lsm_c_dcore_mn, lsm_c_dcore_sd,
lsm_c_dcore_cv, lsm_l_dcore_mn, lsm_l_dcore_cv

## Examples

```
lsm_l_dcore_sd(landscape)
```

---

# lsm_l_division *DIVISION (landscape level)*

---

## Description

Landscape division index (Aggregation metric)

## Usage

```
lsm_l_division(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_division(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_division(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_division(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_division(landscape, directions = 8)
```

```
## S3 method for class 'list'
lsm_l_division(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$DIVISON = (1 - \sum_{i=1}^{m} \sum_{j=1}^{n} (\frac{a_{ij}}{A})^2)$$

where $a_{ij}$ is the area in square meters and $A$ is the total landscape area in square meters.

DIVISION is an 'Aggregation metric. It can be in as the probability that two randomly selected cells are not located in the same patch. The landscape division index is negatively correlated with the effective mesh size (`lsm_c_mesh`).

**Units:** Proportion

**Ranges:** 0 <= Division < 1

**Behaviour:** Equals DIVISION = 0 if only one patch is present. Approaches DIVISION = 1 if all patches of class i are single cells.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Jaeger, J. A. 2000. Landscape division, splitting index, and effective mesh size: new measures of landscape fragmentation. Landscape ecology, 15(2), 115-130.

## See Also

`lsm_p_area`, `lsm_l_ta`,
`lsm_c_division`

## Examples

```
lsm_l_division(landscape)
```

## lsm_l_ed *ED (landscape level)*

### Description

Edge Density (Area and Edge metric)

### Usage

```
lsm_l_ed(landscape, count_boundary, directions)

## S3 method for class 'RasterLayer'
lsm_l_ed(landscape, count_boundary = FALSE, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_ed(landscape, count_boundary = FALSE, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_ed(landscape, count_boundary = FALSE, directions = 8)

## S3 method for class 'stars'
lsm_l_ed(landscape, count_boundary = FALSE, directions = 8)

## S3 method for class 'list'
lsm_l_ed(landscape, count_boundary = FALSE, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `count_boundary` | Count landscape boundary as edge |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$ED = \frac{E}{A} * 10000$$

where $E$ is the total landscape edge in meters and $A$ is the total landscape area in square meters.

ED is an 'Area and Edge metric'. The edge density equals all edges in the landscape in relation to the landscape area. The boundary of the landscape is only included in the corresponding total class edge length if `count_boundary = TRUE`. The metric describes the configuration of the landscape, e.g. because an overall aggregation of classes will result in a low edge density. The metric is standardized to the total landscape area, and therefore comparisons among landscapes with different total areas are possible.

**Units:** Meters per hectare

**Range:** ED >= 0

**Behaviour:** Equals ED = 0 if only one patch is present (and the landscape boundary is not included) and increases, without limit, as the landscapes becomes more patchy

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_l_te, lsm_l_ta,
lsm_c_ed

## Examples

```
lsm_l_ed(landscape)
```

---

# lsm_l_enn_cv *ENN_CV (landscape level)*

---

## Description

Coefficient of variation of euclidean nearest-neighbor distance (Aggregation metric)

## Usage

```
lsm_l_enn_cv(landscape, directions, verbose)

## S3 method for class 'RasterLayer'
lsm_l_enn_cv(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_l_enn_cv(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_l_enn_cv(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'stars'
lsm_l_enn_cv(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_l_enn_cv(landscape, directions = 8, verbose = TRUE)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `verbose` | Print warning message if not sufficient patches are present |

## Details

$$ENN_{CV} = cv(ENN[patch_{ij}])$$

where $ENN[patch_{ij}]$ is the euclidean nearest-neighbor distance of each patch.

ENN_CV is an 'Aggregation metric'. It summarises the landscape as the Coefficient of variation of all patches in the landscape. ENN measures the distance to the nearest neighbouring patch of the same class i. The distance is measured from edge-to-edge. The range is limited by the cell resolution on the lower limit and the landscape extent on the upper limit. The metric is a simple way to describe patch isolation. Because it is scaled to the mean, it is easily comparable among different landscapes.

**Units:** Meters

**Range:** ENN_CV >= 0

**Behaviour:** Equals ENN_CV = 0 if the euclidean nearest-neighbor distance is identical for all patches. Increases, without limit, as the variation of ENN increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

McGarigal, K., and McComb, W. C. (1995). Relationships between landscape structure and breeding birds in the Oregon Coast Range. Ecological monographs, 65(3), 235-260.

## See Also

`lsm_p_enn`, `cv`,
`lsm_c_enn_mn`, `lsm_c_enn_sd`, `lsm_c_enn_cv`,
`lsm_l_enn_mn`, `lsm_l_enn_sd`,

## Examples

```
lsm_l_enn_cv(landscape)
```

## lsm_l_enn_mn *ENN_MN (landscape level)*

### Description

Mean of euclidean nearest-neighbor distance (Aggregation metric)

### Usage

```
lsm_l_enn_mn(landscape, directions, verbose)

## S3 method for class 'RasterLayer'
lsm_l_enn_mn(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_l_enn_mn(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_l_enn_mn(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'stars'
lsm_l_enn_mn(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_l_enn_mn(landscape, directions = 8, verbose = TRUE)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `verbose` | Print warning message if not sufficient patches are present |

### Details

$$ENN_{MN} = cv(mean[patch_{ij}])$$

where $ENN[patch_{ij}]$ is the euclidean nearest-neighbor distance of each patch.

ENN_CV is an 'Aggregation metric'. It summarises the landscape as the mean of all patches in the landscape. ENN measures the distance to the nearest neighbouring patch of the same class i. The distance is measured from edge-to-edge. The range is limited by the cell resolution on the lower limit and the landscape extent on the upper limit.

**Units:** Meters

**Range:** ENN_MN > 0

**Behaviour:** Approaches ENN_MN = 0 as the distance to the nearest neighbour decreases, i.e. patches of the same class i are more aggregated. Increases, without limit, as the distance between neighbouring patches of the same class i increases, i.e. patches are more isolated.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

McGarigal, K., and McComb, W. C. (1995). Relationships between landscape structure and breeding birds in the Oregon Coast Range. Ecological monographs, 65(3), 235-260.

### See Also

lsm_p_enn, mean,
lsm_c_enn_mn, lsm_c_enn_sd, lsm_c_enn_cv,
lsm_l_enn_sd, lsm_l_enn_cv

### Examples

```
lsm_l_enn_mn(landscape)
```

---

## lsm_l_enn_sd *ENN_SD (landscape level)*

---

### Description

Standard deviation of euclidean nearest-neighbor distance (Aggregation metric)

### Usage

```
lsm_l_enn_sd(landscape, directions, verbose)

## S3 method for class 'RasterLayer'
lsm_l_enn_sd(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_l_enn_sd(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_l_enn_sd(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'stars'
```

```
lsm_l_enn_sd(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_l_enn_sd(landscape, directions = 8, verbose = TRUE)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `verbose` | Print warning message if not sufficient patches are present |

## Details

$$ENN_{SD} = sd(ENN[patch_{ij}])$$

where $ENN[patch_{ij}]$ is the euclidean nearest-neighbor distance of each patch.

ENN_CV is an 'Aggregation metric'. It summarises in the landscape as the standard deviation of all patches in the landscape. ENN measures the distance to the nearest neighbouring patch of the same class i. The distance is measured from edge-to-edge. The range is limited by the cell resolution on the lower limit and the landscape extent on the upper limit. The metric is a simple way to describe patch isolation. Because it is scaled to the mean, it is easily comparable among different landscapes.

**Units:** Meters

**Range:** ENN_SD >= 0

**Behaviour:** Equals ENN_SD = 0 if the euclidean nearest-neighbor distance is identical for all patches. Increases, without limit, as the variation of ENN increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

McGarigal, K., and McComb, W. C. (1995). Relationships between landscape structure and breeding birds in the Oregon Coast Range. Ecological monographs, 65(3), 235-260.

## See Also

`lsm_p_enn`, `sd`,
`lsm_c_enn_mn`, `lsm_c_enn_sd`, `lsm_c_enn_cv`,
`lsm_l_enn_mn`, `lsm_l_enn_cv`

## Examples

```
lsm_l_enn_sd(landscape)
```

---

lsm_l_ent *ENT (landscape level)*

---

## Description

Marginal entropy \[H(x)\]

## Usage

```
lsm_l_ent(landscape, neighbourhood, base)

## S3 method for class 'RasterLayer'
lsm_l_ent(landscape, neighbourhood = 4, base = "log2")

## S3 method for class 'RasterStack'
lsm_l_ent(landscape, neighbourhood = 4, base = "log2")

## S3 method for class 'RasterBrick'
lsm_l_ent(landscape, neighbourhood = 4, base = "log2")

## S3 method for class 'stars'
lsm_l_ent(landscape, neighbourhood = 4, base = "log2")

## S3 method for class 'list'
lsm_l_ent(landscape, neighbourhood = 4, base = "log2")
```

## Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| neighbourhood | The number of directions in which cell adjacencies are considered as neighbours: 4 (rook's case) or 8 (queen's case). The default is 4. |
| base | The unit in which entropy is measured. The default is "log2", which compute entropy in "bits". "log" and "log10" can be also used. |

## Details

It measures a diversity (thematic complexity) of landscape classes.

## Value

tibble

### References

Nowosad J., TF Stepinski. 2019. Information theory as a consistent framework for quantification and classification of landscape patterns. https://doi.org/10.1007/s10980-019-00830-x

### See Also

lsm_l_condent, lsm_l_mutinf, lsm_l_joinent,

### Examples

```
lsm_l_ent(landscape)
```

## lsm_l_frac_cv *FRAC_CV (landscape level)*

### Description

Coefficient of variation fractal dimension index (Shape metric)

### Usage

```
lsm_l_frac_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_frac_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_frac_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_frac_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_frac_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_frac_cv(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$FRAC_{CV} = cv(FRAC[patch_{ij}])$$

where $FRAC[patch_{ij}]$ equals the fractal dimension index of each patch.

FRAC_CV is a 'Shape metric'. The metric summarises the landscape as the Coefficient of variation of the fractal dimension index of all patches in the landscape. The fractal dimension index is based on the patch perimeter and the patch area and describes the patch complexity. The Coefficient of variation is scaled to the mean and comparable among different landscapes.

**Units:** None

**Range:** FRAC_CV >= 0

**Behaviour:** Equals FRAC_CV = 0 if the fractal dimension index is identical for all patches. Increases, without limit, as the variation of the fractal dimension indices increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Mandelbrot, B. B. 1977. Fractals: Form, Chance, and Dimension. San Francisco. W. H. Freeman and Company.

## See Also

lsm_p_frac, cv,
lsm_c_frac_mn, lsm_c_frac_sd, lsm_c_frac_cv,
lsm_l_frac_mn, lsm_l_frac_sd,

## Examples

```
lsm_l_frac_cv(landscape)
```

## lsm_l_frac_mn *FRAC_MN (landscape level)*

### Description

Mean fractal dimension index (Shape metric)

### Usage

```
lsm_l_frac_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_frac_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_frac_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_frac_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_frac_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_frac_mn(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$FRAC_{MN} = mean(FRAC[patch_{ij}])$$

where $FRAC[patch_{ij}]$ equals the fractal dimension index of each patch.

FRAC_MN is a 'Shape metric'. The metric summarises the landscape as the mean of the fractal dimension index of all patches in the landscape. The fractal dimension index is based on the patch perimeter and the patch area and describes the patch complexity. The Coefficient of variation is scaled to the mean and comparable among different landscapes.

**Units:** None

**Range:** FRAC_MN > 0

**Behaviour:** Approaches FRAC_MN = 1 if all patches are squared and FRAC_MN = 2 if all patches are irregular.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Mandelbrot, B. B. 1977. Fractals: Form, Chance, and Dimension. San Francisco. W. H. Freeman and Company.

## See Also

lsm_p_frac, mean,
lsm_c_frac_mn, lsm_c_frac_sd, lsm_c_frac_cv,
lsm_l_frac_sd, lsm_l_frac_cv

## Examples

```
lsm_l_frac_mn(landscape)
```

---

# lsm_l_frac_sd *FRAC_SD (landscape level)*

---

## Description

Standard deviation fractal dimension index (Shape metric)

## Usage

```
lsm_l_frac_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_frac_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_frac_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_frac_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_frac_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_frac_sd(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$FRAC_{SD} = sd(FRAC[patch_{ij}])$$

where $FRAC[patch_{ij}]$ equals the fractal dimension index of each patch.

FRAC_SD is a 'Shape metric'. The metric summarises the landscape as the standard deviation of the fractal dimension index of all patches in the landscape. The fractal dimension index is based on the patch perimeter and the patch area and describes the patch complexity.

**Units:** None

**Range:** FRAC_SD>= 0

**Behaviour:** Equals FRAC_SD = 0 if the fractal dimension index is identical for all patches. Increases, without limit, as the variation of the fractal dimension indices increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Mandelbrot, B. B. 1977. Fractals: Form, Chance, and Dimension. San Francisco. W. H. Freeman and Company.

## See Also

`lsm_p_frac`, `sd`,
`lsm_c_frac_mn`, `lsm_c_frac_sd`, `lsm_c_frac_cv`,
`lsm_l_frac_mn`, `lsm_l_frac_cv`

## Examples

```
lsm_l_frac_sd(landscape)
```

## lsm_l_gyrate_cv *GYRATE_CV (landscape level)*

### Description

Coefficient of variation radius of gyration (Area and edge metric)

### Usage

```
lsm_l_gyrate_cv(landscape, directions, cell_center)

## S3 method for class 'RasterLayer'
lsm_l_gyrate_cv(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'RasterStack'
lsm_l_gyrate_cv(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'RasterBrick'
lsm_l_gyrate_cv(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'stars'
lsm_l_gyrate_cv(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'list'
lsm_l_gyrate_cv(landscape, directions = 8, cell_center = FALSE)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `cell_center` | If true, the coordinates of the centroid are forced to be a cell center within the patch. |

### Details

$$GYRATE_{CV} = cv(GYRATE[patch_{ij}])$$

where $GYRATE[patch_{ij}]$ equals the radius of gyration of each patch.

GYRATE_CV is an 'Area and edge metric'. The metric summarises the landscape as the Coefficient of variation of the radius of gyration of all patches in the landscape. GYRATE measures the distance from each cell to the patch centroid and is based on cell center-to-cell center distances. The metrics characterises both the patch area and compactness. The Coefficient of variation is scaled to the mean and comparable among different landscapes.

If `cell_center = TRUE` some patches might have several possible cell-center centroids. In this case, the gyrate index is based on the mean distance of all cells to all possible cell-center centroids.

**Units:** Meters

**Range:** GYRATE_CV >= 0

**Behaviour:** Equals GYRATE_CV = 0 if the radius of gyration is identical for all patches. Increases, without limit, as the variation of the radius of gyration increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Keitt, T. H., Urban, D. L., & Milne, B. T. 1997. Detecting critical scales in fragmented landscapes. Conservation ecology, 1(1).

## See Also

lsm_p_gyrate, cv,
lsm_c_gyrate_mn, lsm_c_gyrate_sd, lsm_c_gyrate_cv,
lsm_l_gyrate_mn, lsm_l_gyrate_sd

## Examples

```
lsm_l_gyrate_cv(landscape)
```

---

# lsm_l_gyrate_mn *GYRATE_MN (landscape level)*

---

## Description

Mean radius of gyration (Area and edge metric)

## Usage

```
lsm_l_gyrate_mn(landscape, directions, cell_center)

## S3 method for class 'RasterLayer'
lsm_l_gyrate_mn(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'RasterStack'
lsm_l_gyrate_mn(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'RasterBrick'
```

```
lsm_l_gyrate_mn(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'stars'
lsm_l_gyrate_mn(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'list'
lsm_l_gyrate_mn(landscape, directions = 8, cell_center = FALSE)
```

## Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| cell_center | If true, the coordinates of the centroid are forced to be a cell center within the patch. |

## Details

$$GYRATE_{MN} = mean(GYRATE[patch_{ij}])$$

where $GYRATE[patch_{ij}]$ equals the radius of gyration of each patch.

GYRATE_MN is an 'Area and edge metric'. The metric summarises the landscape as the mean of the radius of gyration of all patches in the landscape. GYRATE measures the distance from each cell to the patch centroid and is based on cell center-to-cell center distances. The metrics characterises both the patch area and compactness.

If `cell_center = TRUE` some patches might have several possible cell-center centroids. In this case, the gyrate index is based on the mean distance of all cells to all possible cell-center centroids.

**Units:** Meters

**Range:** GYRATE_MN >= 0

**Behaviour:** Approaches GYRATE_MN = 0 if every patch is a single cell. Increases, without limit, when only one patch is present.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Keitt, T. H., Urban, D. L., & Milne, B. T. 1997. Detecting critical scales in fragmented landscapes. Conservation ecology, 1(1).

## See Also

lsm_p_gyrate, mean,
lsm_c_gyrate_mn, lsm_c_gyrate_sd, lsm_c_gyrate_cv,
lsm_l_gyrate_sd, lsm_l_gyrate_cv

## Examples

```
lsm_l_gyrate_mn(landscape)
```

---

# lsm_l_gyrate_sd *GYRATE_SD (landscape level)*

---

## Description

Standard deviation radius of gyration (Area and edge metric)

## Usage

```
lsm_l_gyrate_sd(landscape, directions, cell_center)

## S3 method for class 'RasterLayer'
lsm_l_gyrate_sd(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'RasterStack'
lsm_l_gyrate_sd(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'RasterBrick'
lsm_l_gyrate_sd(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'stars'
lsm_l_gyrate_sd(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'list'
lsm_l_gyrate_sd(landscape, directions = 8, cell_center = FALSE)
```

## Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| cell_center | If true, the coordinates of the centroid are forced to be a cell center within the patch. |

## Details

$$GYRATE_{SD} = sd(GYRATE[patch_{ij}])$$

where $GYRATE[patch_{ij}]$ equals the radius of gyration of each patch.

GYRATE_SD is an 'Area and edge metric'. The metric summarises the landscape as the standard deviation of the radius of gyration of all patches in the landscape. GYRATE measures the distance from each cell to the patch centroid and is based on cell center-to-cell center distances. The metrics characterises both the patch area and compactness.

If `cell_center` = `TRUE` some patches might have several possible cell-center centroids. In this case, the gyrate index is based on the mean distance of all cells to all possible cell-center centroids.

**Units:** Meters

**Range:** GYRATE_SD >= 0

**Behaviour:** Equals GYRATE_SD = 0 if the radius of gyration is identical for all patches. Increases, without limit, as the variation of the radius of gyration increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Keitt, T. H., Urban, D. L., & Milne, B. T. 1997. Detecting critical scales in fragmented landscapes. Conservation ecology, 1(1).

## See Also

`lsm_p_gyrate`, `cv`,
`lsm_c_gyrate_mn`, `lsm_c_gyrate_sd`, `lsm_c_gyrate_cv`,
`lsm_l_gyrate_mn`, `lsm_l_gyrate_cv`

## Examples

```
lsm_l_gyrate_sd(landscape)
```

## lsm_l_iji *Interspersion and Juxtaposition index (landscape level)*

### Description

Interspersion and Juxtaposition index (Aggregation metric)

### Usage

```
lsm_l_iji(landscape, verbose)

## S3 method for class 'RasterLayer'
lsm_l_iji(landscape, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_l_iji(landscape, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_l_iji(landscape, verbose = TRUE)

## S3 method for class 'stars'
lsm_l_iji(landscape, verbose = TRUE)

## S3 method for class 'list'
lsm_l_iji(landscape, verbose = TRUE)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `verbose` | Print warning message if not sufficient patches are present |

### Details

$$IJI = \frac{- \sum\limits_{i=1}^{m} \sum\limits_{k=i+1}^{m} \left[ \left( \frac{e_{ik}}{E} \right) ln \left( \frac{e_{ik}}{E} \right) \right]}{ln(0.5[m(m-1)])} * 100$$

where $e_{ik}$ are the unique adjacencies of all classes (lower/upper triangle of the adjacency table - without the diagonal), $E$ is the total length of edges in the landscape and $m$ is the number of classes.

IJI is an 'Aggregation metric'. It is a so called "salt and pepper" metric and describes the intermixing of classes (i.e. without considering like adjacencies - the diagonal of the adjacency table). The number of classes to calculate IJI must be >= than 3.

**Units:** Percent

**Range:** 0 < IJI <= 100

**Behaviour:** Approaches 0 if a class is only adjacent to a single other class and equals 100 when a class is equally adjacent to all other classes.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

McGarigal, K., & Marks, B. J. 1995. FRAGSTATS: spatial pattern analysis program for quantifying landscape structure. Gen. Tech. Rep. PNW-GTR-351. Portland, OR: US Department of Agriculture, Forest Service, Pacific Northwest Research Station. 122 p, 351.

### See Also

`lsm_c_iji`

### Examples

```
lsm_l_iji(landscape)
```

---

## `lsm_l_joinent` *JOINENT (landscape level)*

---

### Description

Joint entropy \[H(x, y)\]

### Usage

```
lsm_l_joinent(landscape, neighbourhood, ordered, base)

## S3 method for class 'RasterLayer'
lsm_l_joinent(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")

## S3 method for class 'RasterStack'
lsm_l_joinent(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")

## S3 method for class 'RasterBrick'
lsm_l_joinent(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")

## S3 method for class 'stars'
lsm_l_joinent(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")
```

```
## S3 method for class 'list'
lsm_l_joinent(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `neighbourhood` | The number of directions in which cell adjacencies are considered as neighbours: 4 (rook's case) or 8 (queen's case). The default is 4. |
| `ordered` | The type of pairs considered. Either ordered (TRUE) or unordered (FALSE). The default is TRUE. |
| `base` | The unit in which entropy is measured. The default is "log2", which compute entropy in "bits". "log" and "log10" can be also used. |

## Details

Complexity of a landscape pattern. An overall spatio-thematic complexity metric.

## Value

tibble

## References

Nowosad J., TF Stepinski. 2019. Information theory as a consistent framework for quantification and classification of landscape patterns. https://doi.org/10.1007/s10980-019-00830-x

## See Also

`lsm_l_ent`, `lsm_l_condent`, `lsm_l_mutinf`

## Examples

```
lsm_l_joinent(landscape)
```

---

# `lsm_l_lpi` *LPI (landscape level)*

---

## Description

Largest patch index (Area and Edge metric)

## Usage

```
lsm_l_lpi(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_lpi(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_lpi(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_lpi(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_lpi(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_lpi(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$LPI = \frac{\max(a_{ij})}{A} * 100$$

where $max(a_{ij})$ is the area of the patch in square meters and $A$ is the total landscape area in square meters.

The largest patch index is an 'Area and edge metric'. It is the percentage of the landscape covered by the largest patch in the landscape. It is a simple measure of dominance.

**Units:** Percentage

**Range:** 0 < LPI <= 100

**Behaviour:** Approaches LPI = 0 when the largest patch is becoming small and equals LPI = 100 when only one patch is present

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

### See Also

lsm_p_area, lsm_l_ta,
lsm_c_lpi

### Examples

```
lsm_l_lpi(landscape)
```

## lsm_l_lsi *LSI (landscape level)*

### Description

Landscape shape index (Aggregation metric)

### Usage

```
lsm_l_lsi(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_lsi(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_lsi(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_lsi(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_lsi(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_lsi(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$LSI = \frac{E}{\min E}$$

where $E$ is the total edge length in cell surfaces and $\min E$ is the minimum total edge length in cell surfaces

LSI is an 'Aggregation metric'. It is the ratio between the actual landscape edge length and the hypothetical minimum edge length. The minimum edge length equals the edge length if only one patch would be present.

**Units:** None

**Ranges:** LSI >= 1

**Behaviour:** Equals LSI = 1 when only one squared patch is present. Increases, without limit, as the length of the actual edges increases, i.e. the patches become less compact.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

## See Also

lsm_p_shape,
lsm_c_lsi

## Examples

```
lsm_l_lsi(landscape)
```

---

# lsm_l_mesh *MESH (landscape level)*

---

## Description

Effective Mesh Size (Aggregation metric)

## Usage

```
lsm_l_mesh(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_mesh(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_mesh(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_mesh(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_mesh(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_mesh(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$MESH = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2}{A} * \frac{1}{10000}$$

where $a_{ij}$ is the patch area in square meters and $A$ is the total landscape area in square meters.

The effective mesh size is an 'Aggregation metric'. Because each patch is squared before the sum is calculated and the sum is standardized by the total landscape area, MESH is a relative measure of patch structure. MESH is perfectly, negatively correlated to `lsm_c_division`.

**Units:** Hectares

**Range:** cell size / total area <= MESH <= total area

**Behaviour:** Equals cellsize/total area if class covers only one cell and equals total area if only one patch is present.

## Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Jaeger, J. A. 2000. Landscape division, splitting index, and effective mesh size: new measures of landscape fragmentation. Landscape ecology, 15(2), 115-130.

### See Also

lsm_p_area, lsm_l_ta,
lsm_c_mesh

### Examples

```
lsm_l_mesh(landscape)
```

---

## lsm_l_msidi *MSIDI (landscape level)*

---

### Description

Modified Simpson's diversity index (Diversity metric)

### Usage

```
lsm_l_msidi(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_msidi(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_msidi(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_msidi(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_msidi(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_msidi(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$MSIDI = -\ln \sum_{i=1}^{m} P_i^2$$

where $P_i$ is the landscape area proportion of class i.

MSIDI is a 'Diversity metric'.

**Units:** None

**Range:** MSIDI >= 0

**Behaviour:** MSIDI = 0 if only one patch is present and increases, without limit, as the amount of patches with equally distributed landscape proportions increases

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Simpson, E. H. 1949. Measurement of diversity. Nature 163:688

Pielou, E. C. 1975. Ecological Diversity. Wiley-Interscience, New York.

Romme, W. H. 1982. Fire and landscapediversity in subalpine forests of Yellowstone National Park.Ecol.Monogr. 52:199-221

## See Also

lsm_l_sidi

## Examples

```
lsm_l_msidi(landscape)
```

## lsm_l_msiei *MSIEI (landscape level)*

### Description

Modified Simpson's evenness index (Diversity metric)

### Usage

```
lsm_l_msiei(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_msiei(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_msiei(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_msiei(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_msiei(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_msiei(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$MSIEi = \frac{-\ln \sum\limits_{i=1}^{m} P_i^2}{\ln m}$$

where $P_i$ is the landscape area proportion of class i. MSIEI is a 'Diversity metric'.

**Units:** None

**Range:** 0 <= MSIEI < 1

**Behaviour:** MSIEI = 0 when only one patch is present and approaches MSIEI = 1 as the proportional distribution of patches becomes more even

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Simpson, E. H. 1949. Measurement of diversity. Nature 163:688

Pielou, E. C. 1975. Ecological Diversity. Wiley-Interscience, New York.

### See Also

lsm_l_siei

### Examples

```
lsm_l_msiei(landscape)
```

---

## lsm_l_mutinf *MUTINF (landscape level)*

---

### Description

Mutual information \[I(y,x)\]

### Usage

```
lsm_l_mutinf(landscape, neighbourhood, ordered, base)

## S3 method for class 'RasterLayer'
lsm_l_mutinf(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")

## S3 method for class 'RasterStack'
lsm_l_mutinf(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")

## S3 method for class 'RasterBrick'
lsm_l_mutinf(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")

## S3 method for class 'stars'
lsm_l_mutinf(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")

## S3 method for class 'list'
lsm_l_mutinf(landscape, neighbourhood = 4, ordered = TRUE, base = "log2")
```

### Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`neighbourhood` The number of directions in which cell adjacencies are considered as neighbours: 4 (rook's case) or 8 (queen's case). The default is 4.

`ordered` The type of pairs considered. Either ordered (TRUE) or unordered (FALSE). The default is TRUE.

`base` The unit in which entropy is measured. The default is "log2", which compute entropy in "bits". "log" and "log10" can be also used.

### Details

It disambiguates landscape pattern types characterize by the same value of an overall complexity (`lsm_l_joinent`).

### Value

tibble

### References

Nowosad J., TF Stepinski. 2019. Information theory as a consistent framework for quantification and classification of landscape patterns. https://doi.org/10.1007/s10980-019-00830-x

### See Also

`lsm_l_ent`, `lsm_l_condent`, `lsm_l_joinent`,

### Examples

```
lsm_l_mutinf(landscape)
```

## `lsm_l_ndca` *NDCA (landscape level)*

### Description

Number of disjunct core areas (Core area metric)

## Usage

```
lsm_l_ndca(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_ndca(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_l_ndca(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_l_ndca(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_l_ndca(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_l_ndca(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary`
Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$NDCA = \sum_{i=1}^{m} \sum_{j=1}^{n} n_{ij}^{core}$$

where $n_{ij}^{core}$ is the number of disjunct core areas.

NDCA is a 'Core area metric'. The metric summarises the landscape as the sum of all patches in the landscape. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). NDCA counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells. It describes patch area and shape simultaneously (more core area when the patch is large, however, the shape must allow disjunct core areas). Thereby, a compact shape (e.g. a square) will contain less disjunct core areas than a more irregular patch.

**Units:** None

**Range:** NDCA >= 0

**Behaviour:** NDCA = 0 when TCA = 0, i.e. every cell in the landscape is an edge cell. NDCA increases, with out limit, as core area increases and patch shapes allow disjunct core areas (i.e. patch shapes become rather complex).

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_c_tca`,
`lsm_p_ncore`, `lsm_c_ndca`

## Examples

```
lsm_l_ndca(landscape)
```

## lsm_l_np *NP (landscape level)*

### Description

Number of patches (Aggregation metric)

### Usage

```
lsm_l_np(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_np(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_np(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_np(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_np(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_np(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$NP = N$$

where $N$ is the number of patches.

NP is an 'Aggregation metric'. It describes the fragmentation of the landscape, however, does not necessarily contain information about the configuration or composition of the landscape.

**Units:** None

**Ranges:** NP >= 1

**Behaviour:** Equals NP = 1 when only one patch is present and increases, without limit, as the number of patches increases

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

### See Also

`lsm_c_np`

### Examples

```
lsm_l_np(landscape)
```

---

## lsm_l_pafrac *PAFRAC (landscape level)*

---

### Description

Perimeter-Area Fractal Dimension (Shape metric)

### Usage

```
lsm_l_pafrac(landscape, directions, verbose)

## S3 method for class 'RasterLayer'
lsm_l_pafrac(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_l_pafrac(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_l_pafrac(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'stars'
lsm_l_pafrac(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_l_pafrac(landscape, directions = 8, verbose = TRUE)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `verbose` | Print warning message if not sufficient patches are present |

## Details

$$PAFRAC = \frac{2}{\beta}$$

where $\beta$ is the slope of the regression of the area against the perimeter (logarithm) $N \sum_{i=1}^{m} \sum_{j=1}^{n} \ln a_{ij} = a + \beta N \sum_{i=1}^{m} \sum_{j=1}^{n} \ln p_{ij}$

PAFRAC is a 'Shape metric'. It describes the patch complexity of the landscape while being scale independent. This means that increasing the patch size while not changing the patch form will not change the metric. However, it is only meaningful if the relationship between the area and perimeter is linear on a logarithmic scale. Furthermore, if there are less than 10 patches in the landscape, the metric returns NA because of the small-sample issue.

**Units:** None

**Range:** 1 <= PAFRAC <= 2

**Behaviour:** Approaches PAFRAC = 1 for patches with simple shapes and approaches PAFRAC = 2 for irregular shapes

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Burrough, P. A. 1986. Principles of Geographical Information Systems for Land Resources Assessment. Monographs on Soil and Resources Survey No. 12. Clarendon Press, Oxford

## See Also

`lsm_p_area`, `lsm_p_perim`,
`lsm_c_pafrac`

## Examples

```
lsm_l_pafrac(landscape)
```

## lsm_l_para_cv *PARA_CV (landscape level)*

### Description

Coefficient of variation perimeter-area ratio (Shape metric)

### Usage

```
lsm_l_para_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_para_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_para_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_para_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_para_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_para_cv(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$PARA_{CV} = cv(PARA[patch_{ij}]$$

where $PARA[patch_{ij}]$ is the perimeter area ratio of each patch.

PARA_CV is a 'Shape metric'. It summarises the landscape as the Coefficient of variation of each patch belonging in the landscape The perimeter-area ratio describes the patch complexity in a straightforward way. However, because it is not standarised to a certain shape (e.g. a square), it is not scale independent, meaning that increasing the patch size while not changing the patch form will change the ratio.

**Units:** None

**Range:** PARA_CV >= 0

**Behaviour:** Equals PARA_CV = 0 if the perimeter-area ratio is identical for all patches. Increases, without limit, as the variation of the perimeter-area ratio increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_para, cv,
lsm_c_para_mn, lsm_c_para_sd, lsm_c_para_cv,
lsm_l_para_mn, lsm_l_para_sd

## Examples

```
lsm_l_para_cv(landscape)
```

---

# lsm_l_para_mn *PARA_MN (landscape level)*

---

## Description

Mean perimeter-area ratio (Shape metric)

## Usage

```
lsm_l_para_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_para_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_para_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_para_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_para_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_para_mn(landscape, directions = 8)
```

## Arguments

landscape Raster* Layer, Stack, Brick or a list of rasterLayers.

directions The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

## Details

$$PARA_{MN} = mean(PARA[patch_{ij}]$$

where $PARA[patch_{ij}]$ is the perimeter area ratio of each patch.

PARA_MN is a 'Shape metric'. It summarises the landscape as the mean of each patch in the landscape. The perimeter-area ratio describes the patch complexity in a straightforward way. However, because it is not standarised to a certain shape (e.g. a square), it is not scale independent, meaning that increasing the patch size while not changing the patch form will change the ratio.

**Units:** None

**Range:** PARA_MN > 0

**Behaviour:** Approaches PARA_MN > 0 if PARA for each patch approaches PARA > 0, i.e. the form approaches a rather small square. Increases, without limit, as PARA increases, i.e. patches become more complex.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_para, mean,
lsm_c_para_mn, lsm_c_para_sd, lsm_c_para_cv,
lsm_l_para_sd, lsm_l_para_cv

## Examples

```
lsm_l_para_mn(landscape)
```

## lsm_l_para_sd *PARA_SD (landscape level)*

### Description

Standard deviation perimeter-area ratio (Shape metric)

### Usage

```
lsm_l_para_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_para_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_para_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_para_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_para_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_para_sd(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$PARA_{SD} = sd(PARA[patch_{ij}]$$

where $PARA[patch_{ij}]$ is the perimeter area ratio of each patch.

PARA_SD is a 'Shape metric'. It summarises the landscape as the standard deviation of each patch belonging in the landscape. The perimeter-area ratio describes the patch complexity in a straightforward way. However, because it is not standarised to a certain shape (e.g. a square), it is not scale independent, meaning that increasing the patch size while not changing the patch form will change the ratio.

**Units:** None

**Range:** PARA_SD >= 0

**Behaviour:** Equals PARA_SD = 0 if the perimeter-area ratio is identical for all patches. Increases, without limit, as the variation of the perimeter-area ratio increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_para, sd,
lsm_c_para_mn, lsm_c_para_sd, lsm_c_para_cv,
lsm_l_para_mn, lsm_l_para_cv

## Examples

```
lsm_l_para_sd(landscape)
```

---

# lsm_l_pd *PD (landscape level)*

---

## Description

Patch density (Aggregation metric)

## Usage

```
lsm_l_pd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_pd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_pd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_pd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_pd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_pd(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$PD = \frac{N}{A} * 10000 * 100$$

where $N$ is the number of patches and $A$ is the total landscape area in square meters.

PD is an 'Aggregation metric'. It describes the fragmentation the landscape, however, does not necessarily contain information about the configuration or composition of the landscape. In contrast to `lsm_l_np` it is standardized to the area and comparisons among landscapes with different total area are possible.

**Units:** Number per 100 hectares

**Ranges:** 0 < PD <= 1e+06

**Behaviour:** Increases as the landscape gets more patchy. Reaches its maximum if every cell is a different patch.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

### See Also

`lsm_c_np`, `lsm_l_ta`,
`lsm_c_pd`

### Examples

```
lsm_l_pd(landscape)
```

## lsm_l_pladj *PLADJ (landscape level)*

### Description

Percentage of Like Adjacencies (Aggregation metric)

### Usage

```
lsm_l_pladj(landscape)

## S3 method for class 'RasterLayer'
lsm_l_pladj(landscape)

## S3 method for class 'RasterStack'
lsm_l_pladj(landscape)

## S3 method for class 'RasterBrick'
lsm_l_pladj(landscape)

## S3 method for class 'stars'
lsm_l_pladj(landscape)

## S3 method for class 'list'
lsm_l_pladj(landscape)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |

### Details

$$PLADJ = (\frac{g_{ij}}{\sum\limits_{k=1}^{m} g_{ik}}) * 100$$

where $g_{ii}$ is the number of adjacencies between cells of class i and $g_{ik}$ is the number of adjacencies between cells of class i and k.

PLADJ is an 'Aggregation metric'. It calculates the frequency how often patches of different classes i (focal class) and k are next to each other, and following is a measure of class aggregation. The adjacencies are counted using the double-count method.

**Units:** Percent

**Ranges:** 0 <= PLADJ <= 100

**Behaviour:** Equals PLADJ = 0 if class i is maximal disaggregated, i.e. every cell is a different patch. Equals PLADJ = 100 when the only one patch is present.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## Examples

```
lsm_l_pladj(landscape)
```

---

`lsm_l_pr` *PR (landscape level)*

---

## Description

Patch richness (Diversity metric)

## Usage

```
lsm_l_pr(landscape)

## S3 method for class 'RasterLayer'
lsm_l_pr(landscape)

## S3 method for class 'RasterStack'
lsm_l_pr(landscape)

## S3 method for class 'RasterBrick'
lsm_l_pr(landscape)

## S3 method for class 'stars'
lsm_l_pr(landscape)

## S3 method for class 'list'
lsm_l_pr(landscape)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |

## Details

$$PR = m$$

where $m$ is the number of classes

PR is a 'Diversity metric'. It is one of the simplest diversity and composition measures. However, because of its absolute nature, it is not comparable among landscapes with different total areas.

**Units:** None

**Range:** PR >= 1

**Behaviour:** Equals PR = 1 when only one patch is present and increases, without limit, as the number of classes increases

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## Examples

```
lsm_l_pr(landscape)
```

# lsm_l_prd *PRD (landscape level)*

## Description

Patch richness density (Diversity metric)

## Usage

```
lsm_l_prd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_prd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_prd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
```

```
lsm_l_prd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_prd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_prd(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$PRD = \frac{m}{A} * 10000 * 100$$

where $m$ is the number of classes and $A$ is the total landscape area in square meters.

PRD is a 'Diversity metric'. It is one of the simplest diversity and composition measures. In contrast to `lsm_l_pr`, it is a relative measure and following, comparable among landscapes with different total landscape areas.

**Units:** Number per 100 hectares

**Range:** PR > 0

**Behaviour:** Approaches PRD > 1 when only one patch is present and the landscape is rather large. Increases, without limit, as the number of classes increases and the total landscape area decreases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## Examples

```
lsm_l_prd(landscape)
```

## lsm_l_rpr *RPD (landscape level)*

### Description

Relative patch richness (Diversity metric)

### Usage

```
lsm_l_rpr(landscape, classes_max, verbose)

## S3 method for class 'RasterLayer'
lsm_l_rpr(landscape, classes_max = NULL, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_l_rpr(landscape, classes_max = NULL, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_l_rpr(landscape, classes_max = NULL, verbose = TRUE)

## S3 method for class 'stars'
lsm_l_rpr(landscape, classes_max = NULL, verbose = TRUE)

## S3 method for class 'list'
lsm_l_rpr(landscape, classes_max = NULL, verbose = TRUE)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `classes_max` | Potential maximum number of present classes |
| `verbose` | Print warning message if not sufficient patches are present |

### Details

$$RPR = \frac{m}{m_{max}} * 100$$

where $m$ is the number of classes and $m_{max}$ is the (theoretical) maximum number of classes.

RPR is an 'Diversity metric'. The metric calculates the percentage of present classes in the landscape in relation to a (theoretical) number of maximum classes. The user has to specify the maximum number of classes. Note, that if `classes_max` is not provided, the functions returns `NA`.

**Units:** Percentage

**Ranges:** 0 < RPR <= 100

**Behaviour:** Approaches RPR > 0 when only one class type is present, but the maximum number of classes is large. Equals RPR = 100 when m = m_max

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Romme, W. H. 1982. Fire and landscapediversity in subalpine forests of Yellowstone National Park.Ecol.Monogr. 52:199-221

### Examples

```
lsm_l_rpr(landscape, classes_max = 5)
```

---

## lsm_l_shape_cv *SHAPE_CV (landscape level)*

---

### Description

Coefficient of variation shape index (Shape metric)

### Usage

```
lsm_l_shape_cv(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_shape_cv(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_shape_cv(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_shape_cv(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_shape_cv(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_shape_cv(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$SHAPE_{CV} = cv(SHAPE[patch_{ij}])$$

where $SHAPE[patch_{ij}]$ is the shape index of each patch.

SHAPE_CV is a 'Shape metric'. The landscape is summarised as the Coefficient of variation of all patches in the landscape. SHAPE describes the ratio between the actual perimeter of the patch and the hypothetical minimum perimeter of the patch. The minimum perimeter equals the perimeter if the patch would be maximally compact.

**Units:** None

**Range:** SHAPE_CV >= 0

**Behaviour:** Equals SHAPE_CV = 0 if all patches have an identical shape index. Increases, without limit, as the variation of the shape index increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

## See Also

lsm_p_shape, cv,
lsm_c_shape_mn, lsm_c_shape_sd,
lsm_c_shape_cv, lsm_l_shape_mn, lsm_l_shape_sd

## Examples

```
lsm_l_shape_cv(landscape)
```

## lsm_l_shape_mn *SHAPE_MN (landscape level)*

### Description

Mean shape index (Shape metric)

### Usage

```
lsm_l_shape_mn(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_shape_mn(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_shape_mn(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_shape_mn(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_shape_mn(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_shape_mn(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$SHAPE_{MN} = mean(SHAPE[patch_{ij}])$$

where $SHAPE[patch_{ij}]$ is the shape index of each patch.

SHAPE_MN is a 'Shape metric'. The landscape is summarised as the mean of all patches in the landscape. SHAPE describes the ratio between the actual perimeter of the patch and the hypothetical minimum perimeter of the patch. The minimum perimeter equals the perimeter if the patch would be maximally compact.

**Units:** None

**Range:** SHAPE_SD >= 1

**Behaviour:** Equals SHAPE_MN = 0 if all patches are squares. Increases, without limit, as the shapes of patches become more complex.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

## See Also

`lsm_p_shape`, `mean`,
`lsm_c_shape_mn`, `lsm_c_shape_sd`, `lsm_c_shape_cv`,
`lsm_l_shape_sd`, `lsm_l_shape_cv`

## Examples

```
lsm_l_shape_mn(landscape)
```

---

# `lsm_l_shape_sd` *SHAPE_SD (landscape level)*

---

## Description

Standard deviation shape index (Shape metric)

## Usage

```
lsm_l_shape_sd(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_shape_sd(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_shape_sd(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_shape_sd(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_shape_sd(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_shape_sd(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$SHAPE_{SD} = sd(SHAPE[patch_{ij}])$$

where $SHAPE[patch_{ij}]$ is the shape index of each patch.

SHAPE_SD is a 'Shape metric'. The landscape summarised as the standard deviation of all patches in the landscape. SHAPE describes the ratio between the actual perimeter of the patch and the hypothetical minimum perimeter of the patch. The minimum perimeter equals the perimeter if the patch would be maximally compact.

**Units:** None

**Range:** SHAPE_SD >= 0

**Behaviour:** Equals SHAPE_SD = 0 if all patches have an identical shape index. Increases, without limit, as the variation of the shape index increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

## See Also

`lsm_p_shape`, `sd`,
`lsm_c_shape_mn`, `lsm_c_shape_sd`, `lsm_c_shape_cv`,
`lsm_l_shape_mn`, `lsm_l_shape_cv`

## Examples

```
lsm_l_shape_sd(landscape)
```

## lsm_l_shdi *SHDI (landscape level)*

### Description

Shannon's diversity index (Diversity metric)

### Usage

```
lsm_l_shdi(landscape)

## S3 method for class 'RasterLayer'
lsm_l_shdi(landscape)

## S3 method for class 'RasterStack'
lsm_l_shdi(landscape)

## S3 method for class 'RasterBrick'
lsm_l_shdi(landscape)

## S3 method for class 'stars'
lsm_l_shdi(landscape)

## S3 method for class 'list'
lsm_l_shdi(landscape)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |

### Details

$$SHDI = - \sum_{i=1}^{m} (P_i * \ln P_i)$$

where $P_i$ is the proportion of class i.

SHDI is a 'Diversity metric'. It is a widely used metric in biodiversity and ecology and takes both the number of classes and the abundance of each class into account.

**Units:** None

**Range:** SHDI >= 0

**Behaviour:** Equals SHDI = 0 when only one patch is present and increases, without limit, as the number of classes increases while the proportions are equally distributed

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Shannon, C., and W. Weaver. 1949. The mathematical theory of communication. Univ. Illinois-Press, Urbana

### See Also

lsm_c_pland

### Examples

```
lsm_l_shdi(landscape)
```

## lsm_l_shei *SHEI (landscape level)*

### Description

Shannons's evenness index (Diversity metric)

### Usage

```
lsm_l_shei(landscape)

## S3 method for class 'RasterLayer'
lsm_l_shei(landscape)

## S3 method for class 'RasterStack'
lsm_l_shei(landscape)

## S3 method for class 'RasterBrick'
lsm_l_shei(landscape)

## S3 method for class 'stars'
lsm_l_shei(landscape)

## S3 method for class 'list'
lsm_l_shei(landscape)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |

## Details

$$SHEI = \frac{- \sum\limits_{i=1}^{m} (P_{i} * \ln P_{i})}{\ln m}$$

where $P_i$ is the proportion of class i and $m$ is the number of classes.

SHEI is a 'Diversity metric'. It is the ratio between the actual Shannon's diversity index and and the theoretical maximum of the Shannon diversity index. It can be understood as a measure of dominance.

**Units:** None

**Range:** 0 <= SHEI < 1

**Behaviour:** Equals SHEI = 0 when only one patch present and equals SHEI = 1 when the proportion of classes is completely equally distributed

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Shannon, C., and W. Weaver. 1949. The mathematical theory of communication. Univ. Illinois-Press, Urbana

## See Also

lsm_c_pland, lsm_l_pr

## Examples

```
lsm_l_shei(landscape)
```

## lsm_l_sidi *SIDI (landscape level)*

### Description

Simpson's diversity index (Diversity metric)

### Usage

```
lsm_l_sidi(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_sidi(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_sidi(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_sidi(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_sidi(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_sidi(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$SIDI = 1 - \sum_{i=1}^{m} P_i^2$$

where $P_i$ is the proportion of class i and $m$ is the number of classes.

SIDI is a 'Diversity metric'. It is widely used in biodiversity and ecology. It is less sensitive to rare class types than `lsm_l_shdi`. It can be interpreted as the probability that two randomly selected cells belong to the same class.

**Units:** None

**Range:** 0 <= SIDI < 1

**Behaviour:** Equals SIDI = 0 when only one patch is present and approaches SIDI < 1 when the number of class types increases while the proportions are equally distributed

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Simpson, E. H. 1949. Measurement of diversity. Nature 163:688

## See Also

`lsm_c_pland`, `lsm_l_pr`

## Examples

```
lsm_l_sidi(landscape)
```

---

# lsm_l_siei *SIEI (landscape level)*

---

## Description

Simpson's evenness index (Diversity metric)

## Usage

```
lsm_l_siei(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_siei(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_siei(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_siei(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_siei(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_siei(landscape, directions = 8)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

## Details

$$SIEI = \frac{1 - \sum_{i=1}^{m} P_i^2}{1 - \frac{1}{m}}$$

where $P_i$ is the proportion of class i and $m$ is the number of classes.

SIEI is a 'Diversity metric'. The metric is widely used in biodiversity and ecology. It is the ratio between the actual Simpson's diversity index and the theoretical maximum Simpson's diversity index.

**Units:** None

**Range:** 0 < SIEI <= 1

**Behaviour:** Equals SIEI = 0 when only one patch is present and approaches SIEI = 1 when the number of class types increases while the proportions are equally distributed

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Simpson, E. H. 1949. Measurement of diversity. Nature 163:688

## See Also

`lsm_c_pland`, `lsm_l_pr`

## Examples

```
lsm_l_siei(landscape)
```

## `lsm_l_split` *SPLIT (landscape level)*

### Description

Splitting index (Aggregation metric)

### Usage

```
lsm_l_split(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_split(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_split(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_split(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_split(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_split(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$SSPLIT = \frac{A^2}{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2}$$

where $a_{ij}$ is the patch area in square meters and $A$ is the total landscape area.

SPLIT is an 'Aggregation metric'. It describes the number of patches if all patches the landscape would be divided into equally sized patches.

**Units:** None

**Range:** 1 <= SPLIT <= Number of cells squared

**Behaviour:** Equals SPLIT = 1 if only one patch is present. Increases as the number of patches increases and is limited if all cells are a patch

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Jaeger, J. A. 2000. Landscape division, splitting index, and effective mesh size: new measures of landscape fragmentation. Landscape ecology, 15(2), 115-130.

### See Also

`lsm_p_area`, `lsm_l_ta`,
`lsm_c_split`

### Examples

```
lsm_l_split(landscape)
```

## `lsm_l_ta` *TA (landscape level)*

### Description

Total area (Area and edge metric)

### Usage

```
lsm_l_ta(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_l_ta(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_l_ta(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_l_ta(landscape, directions = 8)

## S3 method for class 'stars'
lsm_l_ta(landscape, directions = 8)

## S3 method for class 'list'
lsm_l_ta(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$CA = sum(AREA[patch_{ij}])$$

where $AREA[patch_{ij}]$ is the area of each patch in hectares.

TA is an 'Area and edge metric'. The total (class) area sums the area of all patches in the landscape. It is the area of the observation area.

**Units:** Hectares

**Range:** TA > 0

**Behaviour:** Approaches TA > 0 if the landscape is small and increases, without limit, as the size of the landscape increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_area`, `sum`,
`lsm_c_ca`

## Examples

```
lsm_l_ta(landscape)
```

## lsm_l_tca *TCA (landscape level)*

### Description

Total core area (Core area metric)

### Usage

```
lsm_l_tca(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_l_tca(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_l_tca(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_l_tca(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_l_tca(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_l_tca(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)
```

### Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| consider_boundary | Logical if cells that only neighbour the landscape boundary should be considered as core |
| edge_depth | Distance (in cells) a cell has the be away from the patch edge to be considered as core cell |

### Details

$$TCA = \sum_{j=1}^{n} a_{ij}^{core} * (\frac{1}{10000})$$

where here $a_{ij}^{core}$ is the core area in square meters.

TCA is a 'Core area metric' and equals the sum of core areas of all patches in the landscape. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). In other words, the core area of a patch is all area that is not an edge. It characterises patch areas and shapes of all patches in the landscape simultaneously (more core area when the patch is large and the shape is rather compact, i.e. a square). Additionally, TCA is a measure for the configuration of the landscape, because the sum of edges increase as patches are less aggregated.

**Units:** Hectares

**Range:** TCA >= 0

**Behaviour:** Increases, without limit, as patch areas increase and patch shapes simplify. TCA = 0 when every cell in every patch is an edge.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_core, lsm_c_tca

## Examples

```
lsm_l_tca(landscape)
```

# lsm_l_te *TE (landscape level)*

## Description

Total edge (Area and Edge metric)

## Usage

```
lsm_l_te(landscape, count_boundary)

## S3 method for class 'RasterLayer'
lsm_l_te(landscape, count_boundary = FALSE)

## S3 method for class 'RasterStack'
lsm_l_te(landscape, count_boundary = FALSE)

## S3 method for class 'RasterBrick'
```

```
lsm_l_te(landscape, count_boundary = FALSE)

## S3 method for class 'stars'
lsm_l_te(landscape, count_boundary = FALSE)

## S3 method for class 'list'
lsm_l_te(landscape, count_boundary = FALSE)
```

### Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| count_boundary | Include landscape boundary in edge length |

### Details

$$TE = \sum_{k=1}^{m} e_{ik}$$

where $e_{ik}$ is the edge lengths in meters. TE is an 'Area and edge metric'. Total edge includes all edges. It measures the configuration of the landscape because a highly fragmented landscape will have many edges. However, total edge is an absolute measure, making comparisons among landscapes with different total areas difficult. If cound_boundary = TRUE also edges to the landscape boundary are included.

**Units:** Meters

**Range:** TE >= 0

**Behaviour:** Equals TE = 0 if all cells are edge cells. Increases, without limit, as landscape becomes more fragmented

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

### See Also

lsm_p_perim lsm_l_te

### Examples

```
lsm_l_te(landscape)
```

## lsm_p_area *AREA (patch level)*

### Description

Patch area (Area and edge metric)

### Usage

```
lsm_p_area(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_area(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_p_area(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_p_area(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_area(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_area(landscape, directions = 8)
```

### Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

### Details

$$AREA = a_{ij} * (\frac{1}{10000})$$

where $a_{ij}$ is the area in square meters.

AREA is an 'Area and edge metric' and equals the area of each patch in hectares. The lower limit of AREA is limited by the resolution of the input raster, i.e. AREA can't be smaller than the resolution squared (in hectares). It is one of the most basic, but also most important metrics, to characterise a landscape. The metric is the simplest measure of composition.

**Units:** Hectares

**Range:** AREA > 0

**Behaviour:** Increases, without limit, as the patch size increases.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

### See Also

lsm_c_area_mn, lsm_c_area_sd, lsm_c_area_cv, lsm_c_ca,
lsm_l_area_mn, lsm_l_area_sd, lsm_l_area_cv, lsm_l_ta

### Examples

```
lsm_p_area(landscape)
```

## lsm_p_cai *CAI (patch level)*

### Description

Core area index (Core area metric)

### Usage

```
lsm_p_cai(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_p_cai(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterStack'
lsm_p_cai(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'RasterBrick'
lsm_p_cai(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'stars'
lsm_p_cai(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)

## S3 method for class 'list'
lsm_p_cai(landscape, directions = 8, consider_boundary = FALSE, edge_depth = 1)
```

## Arguments

landscape Raster* Layer, Stack, Brick or a list of rasterLayers.

directions The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

consider_boundary Logical if cells that only neighbour the landscape boundary should be considered as core

edge_depth Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$CAI = (\frac{a_{ij}^{core}}{a_{ij}}) * 100$$

where $a_{ij}^{core}$ is the core area in square meters and $a_{ij}$ is the area in square meters.

CAI is a 'Core area metric'. It equals the percentage of a patch that is core area. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). It describes patch area and shape simultaneously (more core area when the patch is large and the shape is rather compact, i.e. a square). Because the index is relative, it is comparable among patches with different area.

**Units:** Percent

**Range:** 0 <= CAI <= 100

**Behaviour:** CAI = 0 when the patch has no core area and approaches CAI = 100 with increasing percentage of core area within a patch.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_p_core, lsm_p_area,
lsm_c_cai_mn, lsm_c_cai_sd, lsm_c_cai_cv, lsm_c_cpland,
lsm_l_cai_mn, lsm_l_cai_sd, lsm_l_cai_cv

## Examples

```
lsm_p_cai(landscape)
```

lsm_p_circle *CIRCLE (patch level)*

## Description

Related Circumscribing Circle (Shape metric)

## Usage

```
lsm_p_circle(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_circle(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_p_circle(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_p_circle(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_circle(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_circle(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$CIRCLE = 1 - (\frac{a_{ij}}{a_{ij}^{circle}})$$

where $a_{ij}$ is the area in square meters and $a_{ij}^{circle}$ the area of the smallest circumscribing circle.

CIRCLE is a 'Shape metric'. The metric is the ratio between the patch area and the smallest circumscribing circle of the patch. The diameter of the smallest circumscribing circle is the 'diameter' of the patch connecting the opposing corner points of the two cells that are the furthest away from each other. The metric characterises the compactness of the patch and is comparable among patches with different area.

**Units:** None

**Range:** 0 <= CIRCLE < 1

**Behaviour:** CIRCLE = 0 for a circular patch and approaches CIRCLE = 1 for a linear patch.

### Value

tibble

### References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Baker, W. L., and Y. Cai. 1992. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. Landscape Ecology 7: 291-302.

Based on C++ code from Project Nayuki (https://www.nayuki.io/page/smallest-enclosing-circle).

### See Also

lsm_p_area,
lsm_c_circle_mn, lsm_c_circle_sd, lsm_c_circle_cv,
lsm_l_circle_mn, lsm_l_circle_sd, lsm_l_circle_cv

### Examples

```
lsm_p_circle(landscape)
```

---

## lsm_p_contig *CONTIG (patch level)*

---

### Description

Contiguity index (Shape metric)

### Usage

```
lsm_p_contig(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_contig(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_p_contig(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_p_contig(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_contig(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_contig(landscape, directions = 8)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

## Details

$$CONTIG = \frac{\left[ \frac{\sum\limits_{r=1}^{z} c_{ijr}}{a_{ij}} \right] - 1}{v - 1}$$

where $c_{ijr}$ is the contiguity value for pixel r in patch ij, $a_{ij}$ the area of the respective patch (number of cells) and $v$ is the size of the filter matrix (13 in this case).

CONTIG is a 'Shape metric'. It asses the spatial connectedness (contiguity) of cells in patches. CONTIG coerces patch values to a value of 1 and the background to NA. A nine cell focal filter matrix:

```
filter_matrix <- matrix(c(1, 2, 1,
                          2, 1, 2,
                          1, 2, 1), 3, 3, byrow = T)
```

... is then used to weight orthogonally contiguous pixels more heavily than diagonally contiguous pixels. Therefore, larger and more connections between patch cells in the rookie case result in larger contiguity index values.

**Units:** None

**Range:** 0 >= CONTIG <= 1

**Behaviour:** Equals 0 for one-pixel patches and increases to a limit of 1 (fully connected patch).

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

LaGro, J. 1991. Assessing patch shape in landscape mosaics. Photogrammetric Engineering and Remote Sensing, 57(3), 285-293

## See Also

`lsm_c_contig_mn`, `lsm_c_contig_sd`, `lsm_c_contig_cv`,
`lsm_l_contig_mn`, `lsm_l_contig_sd`, `lsm_l_contig_cv`

## Examples

```
lsm_p_contig(landscape)
```

---

`lsm_p_core` *CORE (patch level)*

---

## Description

Core area (Core area metric)

## Usage

```
lsm_p_core(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_p_core(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_p_core(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_p_core(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_p_core(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

```
## S3 method for class 'list'
lsm_p_core(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

## Details

$$CORE = a_{ij}^{core}$$

where $a_{ij}^{core}$ is the core area in square meters

CORE is a 'Core area metric' and equals the area within a patch that is not on the edge of it. A cell is defined as core area if the cell has no neighbour with a different value than itself (rook's case). It describes patch area and shape simultaneously (more core area when the patch is large and the shape is rather compact, i.e. a square).

**Units:** Hectares

**Range:** CORE >= 0

**Behaviour:** Increases, without limit, as the patch area increases and the patch shape simplifies (more core area). CORE = 0 when every cell in the patch is an edge.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_c_core_mn`, `lsm_c_core_sd`, `lsm_c_core_cv`, `lsm_c_tca`,
`lsm_l_core_mn`, `lsm_l_core_sd`, `lsm_l_core_cv`, `lsm_l_tca`

## Examples

```
lsm_p_core(landscape)
```

---

`lsm_p_enn` *ENN (patch level)*

---

## Description

Euclidean Nearest-Neighbor Distance (Aggregation metric)

## Usage

```
lsm_p_enn(landscape, directions, verbose)

## S3 method for class 'RasterLayer'
lsm_p_enn(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterStack'
lsm_p_enn(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'RasterBrick'
lsm_p_enn(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'stars'
lsm_p_enn(landscape, directions = 8, verbose = TRUE)

## S3 method for class 'list'
lsm_p_enn(landscape, directions = 8, verbose = TRUE)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `verbose` | Print warning message if not sufficient patches are present |

## Details

$$ENN = h_{ij}$$

where $h_{ij}$ is the distance to the nearest neighbouring patch of the same class i in meters

ENN is an 'Aggregation metric'. The distance to the nearest neighbouring patch of the same class i. The distance is measured from edge-to-edge. The range is limited by the cell resolution on the lower limit and the landscape extent on the upper limit. The metric is a simple way to describe patch isolation.

**Units:** Meters

**Range:** ENN > 0

**Behaviour:** Approaches ENN = 0 as the distance to the nearest neighbour decreases, i.e. patches of the same class i are more aggregated. Increases, without limit, as the distance between neighbouring patches of the same class i increases, i.e. patches are more isolated.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

McGarigal, K., and McComb, W. C. (1995). Relationships between landscape structure and breeding birds in the Oregon Coast Range. Ecological monographs, 65(3), 235-260.

## See Also

lsm_c_enn_mn, lsm_c_enn_sd, lsm_c_enn_cv,
lsm_l_enn_mn, lsm_l_enn_sd, lsm_l_enn_cv

## Examples

```
lsm_p_enn(landscape)
```

# lsm_p_frac *FRAC (patch level)*

## Description

Fractal dimension index (Shape metric)

## Usage

```
lsm_p_frac(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_frac(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_p_frac(landscape, directions = 8)

## S3 method for class 'RasterBrick'
```

```
lsm_p_frac(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_frac(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_frac(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$FRAC = \frac{2 * \ln * (0.25 * p_{ij})}{\ln a_{ij}}$$

where $p_{ij}$ is the perimeter in meters and $a_{ij}$ is the area in square meters

FRAC is a 'Shape metric'. The index is based on the patch perimeter and the patch area and describes the patch complexity. Because it is standardized, it is scale independent, meaning that increasing the patch size while not changing the patch form will not change the ratio.

**Units:** None

**Range:** 1 <= FRAC <= 2

**Behaviour:** Approaches FRAC = 1 for a squared patch shape form and FRAC = 2 for a irregular patch shape.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Mandelbrot, B. B. 1977. Fractals: Form, Chance, and Dimension. San Francisco. W. H. Freeman and Company.

## See Also

`lsm_p_area`, `lsm_p_perim`,
`lsm_c_frac_mn`, `lsm_c_frac_sd`, `lsm_c_frac_cv`,
`lsm_l_frac_mn`, `lsm_l_frac_sd`, `lsm_l_frac_cv`

## Examples

```
lsm_p_frac(landscape)
```

---

`lsm_p_gyrate` *GYRATE (patch level)*

---

## Description

Radius of Gyration (Area and edge metric)

## Usage

```
lsm_p_gyrate(landscape, directions, cell_center)

## S3 method for class 'RasterLayer'
lsm_p_gyrate(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'RasterStack'
lsm_p_gyrate(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'RasterBrick'
lsm_p_gyrate(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'stars'
lsm_p_gyrate(landscape, directions = 8, cell_center = FALSE)

## S3 method for class 'list'
lsm_p_gyrate(landscape, directions = 8, cell_center = FALSE)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `cell_center` | If true, the coordinates of the centroid are forced to be a cell center within the patch. |

## Details

$$GYRATE = \sum_{r=1}^{z} \frac{h_{ijr}}{z}$$

where $h_{ijr}$ is the distance from each cell to the centroid of the patch and $z$ is the number of cells.

GYRATE is an 'Area and edge metric'. The distance from each cell to the patch centroid is based on cell center to centroid distances. The metric characterises both the patch area and compactness.

If `cell_center` = `TRUE` some patches might have several possible cell-center centroids. In this case, the gyrate index is based on the mean distance of all cells to all possible cell-center centroids.

**Units:** Meters

**Range:** GYRATE >= 0

**Behaviour:** Approaches GYRATE = 0 if patch is a single cell. Increases, without limit, when only one patch is present.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Keitt, T. H., Urban, D. L., & Milne, B. T. 1997. Detecting critical scales in fragmented landscapes. Conservation ecology, 1(1).

## See Also

`lsm_c_gyrate_mn`, `lsm_c_gyrate_sd`, `lsm_c_gyrate_cv`, `lsm_l_gyrate_mn`, `lsm_l_gyrate_sd`, `lsm_l_gyrate_cv`

## Examples

```
lsm_p_gyrate(landscape)
```

---

# `lsm_p_ncore` *NCORE (patch level)*

---

## Description

Number of core areas (Core area metric)

## Usage

```
lsm_p_ncore(landscape, directions, consider_boundary, edge_depth)

## S3 method for class 'RasterLayer'
lsm_p_ncore(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
lsm_p_ncore(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
lsm_p_ncore(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'stars'
lsm_p_ncore(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
lsm_p_ncore(
  landscape,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

### Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

#' @details

$$NCORE = n_{ij}^{core}$$

where $n_{ij}^{core}$ is the number of disjunct core areas.

NCORE is a 'Core area metric'. A cell is defined as core if the cell has no neighbour with a different value than itself (rook's case). The metric counts the disjunct core areas, whereby a core area is a 'patch within the patch' containing only core cells. It describes patch area and shape simultaneously (more core area when the patch is large, however, the shape must allow disjunct core areas). Thereby, a compact shape (e.g. a square) will contain less disjunct core areas than a more irregular patch.

**Units:** None

**Range:** NCORE >= 0

**Behaviour:** NCORE = 0 when CORE = 0, i.e. every cell in patch is edge. Increases, without limit, as core area increases and patch shape allows disjunct core areas (i.e. patch shape becomes rather complex).

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

lsm_c_dcore_mn, lsm_c_dcore_sd, lsm_c_dcore_cv, lsm_c_ndca,
lsm_l_dcore_mn, lsm_l_dcore_sd, lsm_l_dcore_cv, lsm_l_ndca

## Examples

```
lsm_p_ncore(landscape)
```

# lsm_p_para *PARA (patch level)*

## Description

Perimeter-Area ratio (Shape metric)

## Usage

```
lsm_p_para(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_para(landscape, directions = 8)
```

```
## S3 method for class 'RasterStack'
lsm_p_para(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_p_para(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_para(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_para(landscape, directions = 8)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |

## Details

$$PARA = \frac{p_{ij}}{a_{ij}}$$

where $p_{ij}$ is the perimeter in meters and $a_{ij}$ is the area in square meters.

PARA is a 'Shape metric'. It describes the patch complexity in a straightforward way. However, because it is not standarised to a certain shape (e.g. a square), it is not scale independent, meaning that increasing the patch size while not changing the patch form will change the ratio.

**Units:** None

**Range:** PARA > 0

**Behaviour:** Increases, without limit, as the shape complexity increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## See Also

`lsm_p_area`, `lsm_p_perim`,
`lsm_c_para_mn`, `lsm_c_para_sd`, `lsm_c_para_cv`,
`lsm_l_para_mn`, `lsm_l_para_sd`, `lsm_l_para_cv`

## Examples

```
lsm_p_para(landscape)
```

---

## lsm_p_perim *PERIM (patch level)*

---

### Description

Perimeter (Area and edge metric))

### Usage

```
lsm_p_perim(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_perim(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_p_perim(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_p_perim(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_perim(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_perim(landscape, directions = 8)
```

### Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

### Details

$$PERIM = p_{ij}$$

where $p_{ij}$ is the perimeter in meters.

PERIM is an 'Area and edge metric'. It equals the perimeter of the patch including also the edge to the landscape boundary. The metric describes patch area (larger perimeter for larger patches), but also patch shape (large perimeter for irregular shapes).

**Units:** Meters

**Range:** PERIM > 0

**Behaviour:** Increases, without limit, as patch size and complexity increases.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

## Examples

```
lsm_p_perim(landscape)
```

---

`lsm_p_shape` *SHAPE (patch level)*

---

## Description

Shape index (Shape metric)

## Usage

```
lsm_p_shape(landscape, directions)

## S3 method for class 'RasterLayer'
lsm_p_shape(landscape, directions = 8)

## S3 method for class 'RasterStack'
lsm_p_shape(landscape, directions = 8)

## S3 method for class 'RasterBrick'
lsm_p_shape(landscape, directions = 8)

## S3 method for class 'stars'
lsm_p_shape(landscape, directions = 8)

## S3 method for class 'list'
lsm_p_shape(landscape, directions = 8)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

## Details

$$SHAPE = \frac{p_{ij}}{\min p_{ij}}$$

where $p_{ij}$ is the perimeter in terms of cell surfaces and $\min p_{ij}$ is the minimum perimeter of the patch in terms of cell surfaces.

SHAPE is a 'Shape metric'. It describes the ratio between the actual perimeter of the patch and the hypothetical minimum perimeter of the patch. The minimum perimeter equals the perimeter if the patch would be maximally compact.

**Units:** None

**Range:** SHAPE >= 1

**Behaviour:** Equals SHAPE = 1 for a squared patch and increases, without limit, as the patch shape becomes more complex.

## Value

tibble

## References

McGarigal, K., SA Cushman, and E Ene. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www.umass.edu/landeco/research/fragstats/

Patton, D. R. 1975. A diversity index for quantifying habitat "edge". Wildl. Soc.Bull. 3:171-173.

## See Also

`lsm_p_perim`, `lsm_p_area`,
`lsm_c_shape_mn`, `lsm_c_shape_sd`, `lsm_c_shape_cv`,
`lsm_l_shape_mn`, `lsm_l_shape_sd`, `lsm_l_shape_cv`

## Examples

```
lsm_p_shape(landscape)
```

## `options_landscapemetrics` *options_landscapemetrics*

### Description

Sets global options for landscapemetrics

### Usage

```
options_landscapemetrics(to_disk = NULL)
```

### Arguments

| | |
|---|---|
| `to_disk` | Logical argument, if FALSE results of get_patches are hold in memory. If true, get_patches writes temporary files and hence, does not hold everything in memory. Can be set with a global option, e.g. `options(to_disk = TRUE)`. See Details. |

### Details

Landscape metrics rely on the delineation of patches. Hence, `get_patches` is heavily used in **landscapemetrics**. As raster can be quite big, the fact that `get_patches` creates a copy of the raster for each class in a landscape becomes a burden for computer memory. Hence, the argument *to_disk* allows to store the results of the connected labeling algorithm on disk. Furthermore, this option can be set globally, so that every function that internally uses `get_patches` can make use of that.

### Value

Global option to be used internally in the package

## `podlasie_ccilc` *Podlasie ESA CCI LC*

### Description

A real landscape of the Podlasie region in Poland from the ESA CCI Land Cover

### Usage

```
podlasie_ccilc
```

### Format

A raster layer object.

## Source

http://maps.elie.ucl.ac.be/CCI/viewer/

---

`sample_lsm` *sample_lsm*

---

## Description

Sample metrics

## Usage

```
sample_lsm(
  landscape,
  y,
  plot_id,
  shape,
  size,
  all_classes,
  return_raster,
  verbose,
  progress,
  ...
)

## S3 method for class 'RasterLayer'
sample_lsm(
  landscape,
  y,
  plot_id = NULL,
  shape = "square",
  size,
  all_classes = FALSE,
  return_raster = FALSE,
  verbose = TRUE,
  progress = FALSE,
  ...
)

## S3 method for class 'RasterStack'
sample_lsm(
  landscape,
  y,
  plot_id = NULL,
  shape = "square",
  size,
  all_classes = FALSE,
  return_raster = FALSE,
  verbose = TRUE,
  progress = FALSE,
  ...
)

## S3 method for class 'RasterBrick'
sample_lsm(
  landscape,
  y,
  plot_id = NULL,
  shape = "square",
  size,
  all_classes = FALSE,
  return_raster = FALSE,
  verbose = TRUE,
  progress = FALSE,
  ...
)

## S3 method for class 'stars'
sample_lsm(
  landscape,
  y,
  plot_id = NULL,
  shape = "square",
  size,
  all_classes = FALSE,
  return_raster = FALSE,
  verbose = TRUE,
  progress = FALSE,
  ...
)

## S3 method for class 'list'
sample_lsm(
  landscape,
  y,
  plot_id = NULL,
  shape = "square",
  size,
  all_classes = FALSE,
  return_raster = FALSE,
  verbose = TRUE,
  progress = FALSE,
  ...
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`y` 2-column matrix with coordinates, SpatialPoints, SpatialLines, SpatialPolygons, sf points or sf polygons.

`plot_id` Vector with id of sample points. If not provided, sample points will be labelled 1...n.

`shape` String specifying plot shape. Either "circle" or "square"

`size` Approximated size of sample plot. Equals the radius for circles or half of the side-length for squares in mapunits. For lines size equals the width of the buffer.

`all_classes` Logical if NA should be returned for classes not present in some sample plots.

`return_raster` Logical if the clipped raster of the sample plot should be returned

`verbose` Print warning messages.

`progress` Print progress report.

`...` Arguments passed on to `calculate_lsm()`.

## Details

This function samples the selected metrics in a buffer area (sample plot) around sample points, sample lines or within provided SpatialPolygons. The size of the actual sampled landscape can be different to the provided size due to two reasons. Firstly, because clipping raster cells using a circle or a sample plot not directly at a cell center lead to inaccuracies. Secondly, sample plots can exceed the landscape boundary. Therefore, we report the actual clipped sample plot area relative in relation to the theoretical, maximum sample plot area e.g. a sample plot only half within the landscape will have a `percentage_inside = 50`. Please be aware that the output is sligthly different to all other `lsm`-function of `landscapemetrics`.

The metrics can be specified by the arguments `what`, `level`, `metric`, `name` and/or `type` (combinations of different arguments are possible (e.g. level = "class", type = "aggregation metric"). If an argument is not provided, automatically all possibilities are selected. Therefore, to get **all** available metrics, don't specify any of the above arguments.

## Value

tibble

## See Also

`list_lsm`
`calculate_lsm`

## Examples

```
# use a matrix
sample_points <- matrix(c(10, 5, 25, 15, 5, 25), ncol = 2, byrow = TRUE)
sample_lsm(landscape, y = sample_points, size = 15, what = ”lsm_l_np”)

# use sp points
```

```
points_sp <- sp::SpatialPoints(sample_points)
sample_lsm(landscape, y = points_sp, size = 15, what = "lsm_l_np", return_raster = TRUE)

## Not run:
# use lines (works only if rgeos is installed)
x1 <- c(1, 5, 15, 10)
y1 <- c(1, 5, 15, 25)

x2 <- c(10, 25)
y2 <- c(5, 5)

sample_lines <- sp::SpatialLines(list(sp::Lines(list(sp::Line(cbind(x1, y1)),
sp::Line(cbind(x2, y2))), ID = "a")))
sample_lsm(landscape, y = sample_lines, size = 10, what = "lsm_l_np")

# use polygons
poly_1 <-  sp::Polygon(cbind(c(2.5, 2.5, 17.5, 17.5),
                           c(-2.5, 12.5, 12.5, -2.5)))
poly_2 <-  sp::Polygon(cbind(c(7.5, 7.5, 23.5, 23.5),
                           c(-7.5, 23.5, 23.5, -7.5)))
poly_1 <- sp::Polygons(list(poly_1), "p1")
poly_2 <- sp::Polygons(list(poly_2), "p2")
sample_plots <- sp::SpatialPolygons(list(poly_1, poly_2))

sample_lsm(landscape, y = sample_plots, what = "lsm_l_np")

## End(Not run)
```

## scale_sample *scale_sample*

### Description

Metrics on changing sample scale

### Usage

```
scale_sample(landscape, y, shape, size, max_size, verbose, progress, ...)

## S3 method for class 'RasterLayer'
scale_sample(
  landscape,
  y,
  shape = "square",
  size,
  max_size,
  verbose = TRUE,
  progress = FALSE,
  ...
)

## S3 method for class 'RasterStack'
scale_sample(
  landscape,
  y,
  shape = "square",
  size,
  max_size,
  verbose = TRUE,
  progress = FALSE,
  ...
)

## S3 method for class 'RasterBrick'
scale_sample(
  landscape,
  y,
  shape = "square",
  size,
  max_size,
  verbose = TRUE,
  progress = FALSE,
  ...
)

## S3 method for class 'stars'
scale_sample(
  landscape,
  y,
  shape = "square",
  size,
  max_size,
  verbose = TRUE,
  progress = FALSE,
  ...
)

## S3 method for class 'list'
scale_sample(
  landscape,
  y,
  shape = "square",
  size,
  max_size,
  verbose = TRUE,
  progress = FALSE,
  ...
)
```

### Arguments

| | |
|---|---|
| landscape | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| y | 2-column matrix with coordinates or SpatialPoints. |
| shape | String specifying plot shape. Either "circle" or "square" |
| size | Approximated size of sample plot. Equals the radius for circles or half of the side-length for squares in mapunits. For lines size equals the width of the buffer. |
| max_size | Maximum size to which sample plot size is summed up. |
| verbose | Print warning messages. |
| progress | Print progress report. |
| ... | Arguments passed on to calculate_lsm(). |

### Details

This function calculates the selected metrics in subsequential buffers around a/multiple point(s) of interest.

The size of the actual sampled landscape can be different to the provided size due to two reasons. Firstly, because clipping raster cells using a circle or a sample plot not directly at a cell center lead to inaccuracies. Secondly, sample plots can exceed the landscape boundary. Therefore, we report the actual clipped sample plot area relative in relation to the theoretical, maximum sample plot area e.g. a sample plot only half within the landscape will have a percentage_inside = 50. Please be aware that the output is sligthly different to all other lsm-function of landscapemetrics.

The metrics can be specified by the arguments what, level, metric, name and/or type (combinations of different arguments are possible (e.g. level = "class", type = "aggregation metric"). If an argument is not provided, automatically all possibilities are selected. Therefore, to get **all** available metrics, don't specify any of the above arguments.

### Value

tibble

### See Also

list_lsm
calculate_lsm
sample_lsm
construct_buffer

### Examples

```
my_points = matrix(c(1265000, 1250000, 1255000, 1257000),
                   ncol = 2, byrow = TRUE)

scale_sample(landscape = augusta_nlcd, y = my_points,
size = 500, max_size = 5000, what = c("lsm_l_ent", "lsm_l_mutinf"))
```

## scale_window *scale_window*

### Description

Metrics on changing sample scale

### Usage

```
scale_window(
  landscape,
  percentages_col,
  percentages_row,
  what,
  stat,
  verbose,
  progress,
  ...
)

## S3 method for class 'RasterLayer'
scale_window(
  landscape,
  percentages_col = c(2, 4, 8, 16, 32, 64, 100),
  percentages_row = NULL,
  what,
  stat,
  verbose = TRUE,
  progress = FALSE,
  ...
)

## S3 method for class 'RasterStack'
scale_window(
  landscape,
  percentages_col = c(2, 4, 8, 16, 32, 64, 100),
  percentages_row = NULL,
  what,
  stat,
  verbose = TRUE,
  progress = FALSE,
  ...
)
```

```
## S3 method for class 'RasterBrick'
scale_window(
  landscape,
  percentages_col = NULL,
  percentages_row = NULL,
  what,
  stat,
  verbose = TRUE,
  progress = FALSE,
  ...
)

## S3 method for class 'stars'
scale_window(
  landscape,
  percentages_col = NULL,
  percentages_row = NULL,
  what,
  stat,
  verbose = TRUE,
  progress = FALSE,
  ...
)

## S3 method for class 'list'
scale_window(
  landscape,
  percentages_col = NULL,
  percentages_row = NULL,
  what,
  stat,
  verbose = TRUE,
  progress = FALSE,
  ...
)
```

## Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`percentages_col` 2-column matrix with coordinates or SpatialPoints.

`percentages_row` String specifying plot shape. Either "circle" or "square"

`what` Selected level of metrics: either "patch", "class" or "landscape". It is also possible to specify functions as a vector of strings, e.g. `what = c("lsm_l_mutinf","lsm_l_ta")`.

`stat` The function to be applied. See Details

`verbose` If TRUE, warnings are printed.

`progress` Print progress report.

`...` Arguments passed on to `calculate_lsm()`.

## Details

This function calculates the selected metrics in moving windows over the provided landscape.

Please be aware that the output is sligthly different to all other `lsm`-function of `landscapemetrics`.

The metrics can be specified by the arguments `what`, `level`, `metric`, `name` and/or `type` (combinations of different arguments are possible (e.g. level = "class", type = "aggregation metric"). If an argument is not provided, automatically all possibilities are selected. Only metrics on landscape level are supported for this function.

## Value

tibble

## See Also

`list_lsm`
`window_lsm`
`scale_sample`

## Examples

```
## Not run:
percentages_col <- c(2, 4, 8, 16, 32, 64, 100)
percentages_row <- c(2, 4, 8, 16, 32, 64, 100)

what =  c("lsm_l_pr", "lsm_l_joinent")

stat <- "mean"

scale_window(landscape, percentages_col, percentages_row, what, stat)

## End(Not run)
```

# `show_cores` *Show core area*

## Description

Show core area

## Usage

```
show_cores(
  landscape,
  directions,
  class,
  labels,
  nrow,
  ncol,
  consider_boundary,
  edge_depth
)

## S3 method for class 'RasterLayer'
show_cores(
  landscape,
  directions = 8,
  class = "all",
  labels = FALSE,
  nrow = NULL,
  ncol = NULL,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterStack'
show_cores(
  landscape,
  directions = 8,
  class = "all",
  labels = FALSE,
  nrow = NULL,
  ncol = NULL,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'RasterBrick'
show_cores(
  landscape,
  directions = 8,
  class = "all",
  labels = FALSE,
  nrow = NULL,
  ncol = NULL,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

```
## S3 method for class 'stars'
show_cores(
  landscape,
  directions = 8,
  class = "all",
  labels = FALSE,
  nrow = NULL,
  ncol = NULL,
  consider_boundary = FALSE,
  edge_depth = 1
)

## S3 method for class 'list'
show_cores(
  landscape,
  directions = 8,
  class = "all",
  labels = FALSE,
  nrow = NULL,
  ncol = NULL,
  consider_boundary = FALSE,
  edge_depth = 1
)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster object |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `class` | How to show the core area: "global" (single map), "all" (every class as facet), or a vector with the specific classes one wants to show (every selected class as facet). |
| `labels` | Logical flag indicating whether to print or not to print core labels. boundary should be considered as core |
| `nrow, ncol` | Number of rows and columns for the facet. |
| `consider_boundary` | Logical if cells that only neighbour the landscape boundary should be considered as core. |
| `edge_depth` | Distance (in cells) a cell has the be away from the patch edge to be considered as core cell |

## Details

The functions plots the core area of patches labeled with the corresponding patch id. The edges are the grey cells surrounding the patches and are always shown.

## Value

ggplot

## Examples

```
# show "global" core area
show_cores(landscape, class = "global", labels = FALSE)

# show the core area of every class as facet
show_cores(landscape, class = "all", labels = FALSE)

# show only the core area of class 1 and 3
show_cores(landscape, class = c(1, 3), labels = TRUE)
```

---

# `show_correlation` *Show correlation*

---

## Description

Show correlation

## Usage

```
show_correlation(
  data,
  method = "pearson",
  diag = TRUE,
  labels = FALSE,
  vjust = 0,
  text_size = 15
)
```

## Arguments

| | |
|---|---|
| `data` | Tibble with results of as returned by the landscapemetrics package. |
| `method` | Type of correlation. See `link{cor}` for details. |
| `diag` | If FALSE, values on the diagonal will be NA and not plotted. |
| `labels` | If TRUE, the correlation value will be added as text. |
| `vjust` | Will be passed on to ggplot2 as vertical justification of x-axis text. |
| `text_size` | Text size of the plot. |

## Details

The functions calculates the correlation between all metrics. In order to calculate correlations, for the landscape level more than one landscape needs to be present. All input must be structured as returned by the **landscapemetrics** package.

## Value

ggplot

## Examples

```
metrics <- calculate_lsm(landscape, what = c(”patch”, ”class”))
show_correlation(data = metrics, method = ”pearson”)

## Not run:
metrics <- calculate_lsm(landscape, what = c(”patch”, ”class”))#'
correlations <- calculate_correlation(metrics)
show_correlation(data = correlations, method = ”pearson”)


## End(Not run)
```

---

# `show_lsm` *Show landscape metrics*

---

## Description

Show landscape metrics on patch level printed in their corresponding patch.

## Usage

```
show_lsm(
  landscape,
  what,
  class,
  directions,
  consider_boundary,
  edge_depth,
  labels,
  label_lsm,
  nrow,
  ncol
)

## S3 method for class 'RasterLayer'
show_lsm(
  landscape,
  what,
  class = ”global”,
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1,
  labels = FALSE,
  label_lsm = FALSE,
  nrow = NULL,
  ncol = NULL
)

## S3 method for class 'RasterStack'
show_lsm(
  landscape,
  what,
  class = "global",
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1,
  labels = FALSE,
  label_lsm = FALSE,
  nrow = NULL,
  ncol = NULL
)

## S3 method for class 'RasterBrick'
show_lsm(
  landscape,
  what,
  class = "global",
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1,
  labels = FALSE,
  label_lsm = FALSE,
  nrow = NULL,
  ncol = NULL
)

## S3 method for class 'stars'
show_lsm(
  landscape,
  what,
  class = "global",
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1,
  labels = FALSE,
  label_lsm = FALSE,
  nrow = NULL,
  ncol = NULL
)

## S3 method for class 'list'
```

```
show_lsm(
  landscape,
  what,
  class = "global",
  directions = 8,
  consider_boundary = FALSE,
  edge_depth = 1,
  labels = FALSE,
  label_lsm = FALSE,
  nrow = NULL,
  ncol = NULL
)
```

## Arguments

`landscape` *Raster object

`what` Patch level what to plot

`class` How to show the labeled patches: "global" (single map), "all" (every class as facet), or a vector with the specific classes one wants to show (every selected class as facet).

`directions` The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case).

`consider_boundary` Logical if cells that only neighbour the landscape boundary should be considered as core

`edge_depth` Distance (in cells) a cell has the be away from the patch edge to be considered as core cell

`labels` Logical flag indicating whether to print or not to print patch labels.

`label_lsm` If true, the value of the landscape metric is used as label

`nrow, ncol` Number of rows and columns for the facet.

## Details

The function plots all patches with a fill corresponding to the value of the chosen landscape metric on patch level.

## Value

ggplot

## Examples

```
show_lsm(landscape, what = "lsm_p_area", directions = 4)
show_lsm(landscape, what = "lsm_p_shape", class = c(1, 2), label_lsm = TRUE)
show_lsm(landscape, what = "lsm_p_circle", class = 3, labels = TRUE)
```

## show_patches *Show patches*

### Description

Show patches

### Usage

```
show_patches(landscape, class, directions, labels, nrow, ncol)

## S3 method for class 'RasterLayer'
show_patches(
  landscape,
  class = "global",
  directions = 8,
  labels = FALSE,
  nrow = NULL,
  ncol = NULL
)

## S3 method for class 'RasterStack'
show_patches(
  landscape,
  class = "global",
  directions = 8,
  labels = FALSE,
  nrow = NULL,
  ncol = NULL
)

## S3 method for class 'RasterBrick'
show_patches(
  landscape,
  class = "global",
  directions = 8,
  labels = FALSE,
  nrow = NULL,
  ncol = NULL
)

## S3 method for class 'stars'
show_patches(
  landscape,
  class = "global",
  directions = 8,
  labels = FALSE,
  nrow = NULL,
  ncol = NULL
)

## S3 method for class 'list'
show_patches(
  landscape,
  class = "global",
  directions = 8,
  labels = FALSE,
  nrow = NULL,
  ncol = NULL
)
```

### Arguments

| | |
|---|---|
| landscape | *Raster object |
| class | How to show the labeled patches: "global" (single map), "all" (every class as facet), or a vector with the specific classes one wants to show (every selected class as facet). |
| directions | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| labels | Logical flag indicating whether to print or not to print patch labels. |
| nrow, ncol | Number of rows and columns for the facet. |

### Details

The functions plots the landscape with the patches labeled with the corresponding patch id.

### Value

ggplot

### Examples

```
show_patches(landscape)
show_patches(landscape, class = c(1, 2))
show_patches(landscape, class = 3, labels = FALSE)
```

## spatialize_lsm *spatialize_lsm*

### Description

Spatialize landscape metric values

### Usage

```
spatialize_lsm(
  landscape,
  level,
  metric,
  name,
  type,
  what,
  directions,
  progress,
  ...
)

## S3 method for class 'RasterLayer'
spatialize_lsm(
  landscape,
  level = "patch",
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  progress = FALSE,
  ...
)

## S3 method for class 'RasterStack'
spatialize_lsm(
  landscape,
  level = "patch",
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  progress = FALSE,
  ...
)

## S3 method for class 'RasterBrick'
spatialize_lsm(
  landscape,
  level = "patch",
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  progress = FALSE,
  ...
)

## S3 method for class 'stars'
spatialize_lsm(
  landscape,
  level = "patch",
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  progress = FALSE,
  ...
)

## S3 method for class 'list'
spatialize_lsm(
  landscape,
  level = "patch",
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  directions = 8,
  progress = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| `landscape` | Raster* Layer, Stack, Brick or a list of rasterLayers. |
| `level` | Level of metrics. Either 'patch', 'class' or 'landscape' (or vector with combination). |
| `metric` | Abbreviation of metrics (e.g. 'area'). |
| `name` | Full name of metrics (e.g. 'core area') |
| `type` | Type according to FRAGSTATS grouping (e.g. 'aggregation metrics'). |
| `what` | Selected level of metrics: either "patch", "class" or "landscape". It is also possible to specify functions as a vector of strings, e.g. `what = c("lsm_c_ca","lsm_l_ta")`. |
| `directions` | The number of directions in which patches should be connected: 4 (rook's case) or 8 (queen's case). |
| `progress` | Print progress report. |
| `...` | Arguments passed on to `calculate_lsm()`. |

## Details

The functions returns a nested list with RasterLayers. The first level contains each input layer (only one element if RasterLayer was provided). The second level contains a RasterLayer for each selected metric (see list_lsm for details) where each cell has the landscape metric value of the patch it belongs to. Only patch level metrics are allowed.

## Value

list

## See Also

list_lsm
show_lsm

## Examples

```
spatialize_lsm(landscape, what = "lsm_p_area")
```

---

# window_lsm *window_lsm*

---

## Description

Moving window

## Usage

```
window_lsm(landscape, window, level, metric, name, type, what, progress, ...)

## S3 method for class 'RasterLayer'
window_lsm(
  landscape,
  window,
  level = "landscape",
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  progress = FALSE,
  ...
)

## S3 method for class 'RasterStack'
window_lsm(
  landscape,
  window,
  level = ”landscape”,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  progress = FALSE,
  ...
)

## S3 method for class 'RasterBrick'
window_lsm(
  landscape,
  window,
  level = ”landscape”,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  progress = FALSE,
  ...
)

## S3 method for class 'stars'
window_lsm(
  landscape,
  window,
  level = ”landscape”,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  progress = FALSE,
  ...
)

## S3 method for class 'list'
window_lsm(
  landscape,
  window,
  level = ”landscape”,
  metric = NULL,
  name = NULL,
  type = NULL,
  what = NULL,
  progress = FALSE,
  ...
)
```

### Arguments

`landscape` Raster* Layer, Stack, Brick or a list of rasterLayers.

`window` Moving window matrix.

`level` Level of metrics. Either 'patch', 'class' or 'landscape' (or vector with combination).

`metric` Abbreviation of metrics (e.g. 'area').

`name` Full name of metrics (e.g. 'core area')

`type` Type according to FRAGSTATS grouping (e.g. 'aggregation metrics').

`what` Selected level of metrics: either "patch", "class" or "landscape". It is also possible to specify functions as a vector of strings, e.g. `what = c("lsm_c_ca","lsm_l_ta")`.

`progress` Print progress report.

`...` Arguments passed on to `calculate_lsm()`.

### Details

The function calculates for each focal cell the selected landscape metrics (currently only landscape level metrics are allowed) for a local neighbourhood. The neighbourhood can be specified using a matrix. For more details, see `?raster::focal()`. The result will be a `RasterLayer` in which each focal cell includes the value of its neighbourhood and thereby allows to show gradients and variability in the landscape (Hagen-Zanker 2016). To be type stable, the acutally result is always a nested list (first level for `RasterStack` layers, second level for selected landscape metrics).

### Value

list

### References

Fletcher, R., Fortin, M.-J. 2018. Spatial Ecology and Conservation Modeling: Applications with R. Springer International Publishing. 523 pages

Hagen-Zanker, A. (2016). A computational framework for generalized moving windows and its application to landscape pattern analysis. International journal of applied earth observation and geoinformation, 44, 205-216.

McGarigal, K., Cushman, S.A., and Ene E. 2012. FRAGSTATS v4: Spatial Pattern Analysis Program for Categorical and Continuous Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following website: `http://www.umass.edu/landeco/research/fragstats/fragstats.html`

### See Also

`list_lsm`
`calculate_lsm`
`focal`

## Examples

```
## Not run:
window <- matrix(1, nrow = 5,ncol = 5)
window_lsm(landscape, window = window, what = c("lsm_l_pr", "lsm_l_joinent"))
window_lsm(landscape_stack, window = window, what = c("lsm_l_pr", "lsm_l_joinent"))

## End(Not run)
```

# Index

∗ **datasets**
  augusta_nlcd, 6
  fragstats_class_augusta_nlcd, 16
  fragstats_class_landscape, 16
  fragstats_class_podlasie, 17
  fragstats_landscape_augusta_nlcd, 17
  fragstats_landscape_landscape, 18
  fragstats_landscape_podlasie, 18
  fragstats_patch_augusta_nlcd, 19
  fragstats_patch_landscape, 19
  fragstats_patch_podlasie, 20
  landscape, 31
  lsm_abbreviations_names, 34
  podlasie_ccilc, 250

augusta_nlcd, 6

calculate_correlation, 6
calculate_lsm, 7, *15*, *253*, *256*, *272*
check_landscape, 11
construct_buffer, *256*
cv, *37*, *44*, *63*, *73*, *82*, *87*, *92*, *95*, *109*, *118*, *129*, *134*, *154*, *163*, *171*, *177*, *182*, *185*, *204*, *215*

extract_lsm, 12

focal, *272*
fragstats_class_augusta_nlcd, 16
fragstats_class_landscape, 16
fragstats_class_podlasie, 17
fragstats_landscape_augusta_nlcd, 17
fragstats_landscape_landscape, 18
fragstats_landscape_podlasie, 18
fragstats_patch_augusta_nlcd, 19
fragstats_patch_landscape, 19
fragstats_patch_podlasie, 20

get_adjacencies, 20
get_boundaries, 22
get_boundaries_calc *(*get_boundaries*)*, 22
get_centroids, 24
get_circumscribingcircle, 25
get_nearestneighbour, 27
get_patches, *27*, 28
get_unique_values, 30

landscape, 31
landscapemetrics, 32
landscapemetrics-package *(*landscapemetrics*)*, 32
list_lsm, *11*, 33, *253*, *256*, *259*, *270*, *272*
lsm_abbreviations_names, 34
lsm_c_ai, 35, *127*
lsm_c_area_cv, 36, *39*, *40*, *129*, *130*, *132*, *232*
lsm_c_area_mn, *37*, 38, *40*, *129*, *130*, *132*, *232*
lsm_c_area_sd, *37*, *39*, 39, *129*, *130*, *132*, *232*
lsm_c_ca, *38*, 41, *117*, *227*, *232*
lsm_c_cai_cv, 42, *46*, *48*, *134*, *136*, *138*, *233*
lsm_c_cai_mn, *44*, 44, *48*, *134*, *138*, *233*
lsm_c_cai_sd, *44*, *46*, 46, *134*, *136*, *138*, *233*
lsm_c_circle_cv, 48, *51*, *53*, *140*, *141*, *143*, *235*
lsm_c_circle_mn, *50*, 50, *53*, *140*, *141*, *143*, *235*
lsm_c_circle_sd, *50*, *51*, 52, *140*, *141*, *143*, *235*
lsm_c_clumpy, 53
lsm_c_cohesion, 55
lsm_c_contig_cv, 56, *57*, *59*, *61*, *149*, *151*, *152*, *236*
lsm_c_contig_mn, *57*, 58, *61*, *149*, *151*, *152*, *236*
lsm_c_contig_sd, *59*, 59, *149*, *151*, *152*, *236*
lsm_c_core_cv, 61, *65*, *67*, *154*, *156*, *158*, *238*
lsm_c_core_mn, *63*, 63, *67*, *154*, *156*, *158*, *238*
lsm_c_core_sd, *63*, *65*, 65, *154*, *156*, *158*, *238*
lsm_c_cpland, 67, *233*
lsm_c_dcad, 69, *161*

lsm_c_dcore_cv, 72, *76*, *78*, *163*, *165*, *167*, *245*
lsm_c_dcore_mn, *73*, 74, *78*, *163*, *165*, *167*, *245*
lsm_c_dcore_sd, *73*, *76*, 76, *163*, *165*, *167*, *245*
lsm_c_division, 78, *101*, *168*, *192*
lsm_c_division.stars (lsm_c_area_sd), 39
lsm_c_ed, 80, *170*
lsm_c_enn_cv, 81, *84*, *86*, *171*, *173*, *174*, *240*
lsm_c_enn_mn, *82*, 83, *86*, *171*, *173*, *174*, *240*
lsm_c_enn_sd, *82*, *84*, 84, *171*, *173*, *174*, *240*
lsm_c_frac_cv, 86, *89*, *90*, *177*, *179*, *180*, *241*
lsm_c_frac_mn, *87*, 88, *90*, *177*, *179*, *180*, *241*
lsm_c_frac_sd, *87*, *89*, 89, *177*, *179*, *180*, *241*
lsm_c_gyrate_cv, 91, *94*, *95*, *182*, *184*, *185*, *243*
lsm_c_gyrate_mn, *92*, 92, *95*, *182*, *184*, *185*, *243*
lsm_c_gyrate_sd, *92*, *94*, 94, *182*, *184*, *185*, *243*
lsm_c_iji, 96, *187*
lsm_c_lpi, 97, *190*
lsm_c_lsi, 99, *191*
lsm_c_mesh, *79*, 100, *168*, *193*
lsm_c_ndca, *71*, 102, *161*, *199*, *245*
lsm_c_nlsi, 104
lsm_c_np, 105, *113*, *114*, *201*, *208*
lsm_c_pafrac, 107, *202*
lsm_c_para_cv, 108, *111*, *112*, *204*, *205*, *207*, *246*
lsm_c_para_mn, *109*, 110, *112*, *204*, *205*, *207*, *246*
lsm_c_para_sd, *109*, *111*, 111, *204*, *205*, *207*, *246*
lsm_c_pd, 113, *208*
lsm_c_pladj, 114
lsm_c_pland, 115, *220*, *221*, *223*, *224*
lsm_c_shape_cv, 117, *120*, *121*, *215*, *217*, *218*, *249*
lsm_c_shape_mn, *118*, 119, *121*, *215*, *217*, *218*, *249*
lsm_c_shape_sd, *118*, *120*, 120, *215*, *217*, *218*, *249*
lsm_c_split, 122, *226*
lsm_c_tca, *104*, 123, *199*, *229*, *238*
lsm_c_te, *81*, 125
lsm_l_ai, *36*, 126
lsm_l_area_cv, *37*, *39*, *40*, 128, *130*, *132*, *232*
lsm_l_area_mn, *37*, *39*, *40*, *129*, 129, *132*, *232*
lsm_l_area_sd, *37*, *39*, *40*, *129*, *130*, 131, *232*
lsm_l_cai_cv, *44*, *46*, *48*, 132, *136*, *138*, *233*
lsm_l_cai_mn, *44*, *46*, *48*, *134*, 134, *138*, *233*
lsm_l_cai_sd, *44*, *46*, *48*, *134*, *136*, 136, *233*
lsm_l_circle_cv, *50*, *51*, *53*, 138, *141*, *143*, *235*
lsm_l_circle_mn, *50*, *51*, *53*, *140*, 140, *143*, *235*
lsm_l_circle_sd, *50*, *51*, *53*, *140*, *141*, 142, *235*
lsm_l_cohesion, *56*, 143, *144*
lsm_l_condent, 145, *176*, *188*, *197*
lsm_l_contag, 146
lsm_l_contig_cv, *57*, *59*, *61*, 148, *151*, *152*, *236*
lsm_l_contig_mn, *57*, *59*, *61*, *149*, 149, *152*, *236*
lsm_l_contig_sd, *57*, *59*, *61*, *149*, *151*, 151, *236*
lsm_l_core_cv, *63*, *65*, *67*, 153, *156*, *158*, *238*
lsm_l_core_mn, *63*, *65*, *67*, *154*, 155, *158*, *238*
lsm_l_core_sd, *63*, *65*, *67*, *154*, *156*, 157, *238*
lsm_l_dcad, *71*, 159
lsm_l_dcore_cv, *73*, *76*, *78*, 161, *165*, *167*, *245*
lsm_l_dcore_mn, *73*, *76*, *78*, *163*, 163, *167*, *245*
lsm_l_dcore_sd, *73*, *76*, *78*, *163*, *165*, 165, *245*
lsm_l_division, *79*, 167
lsm_l_ed, *81*, 169
lsm_l_enn_cv, *82*, *84*, *86*, 170, *173*, *174*, *240*
lsm_l_enn_mn, *82*, *84*, *86*, *171*, 172, *174*, *240*
lsm_l_enn_sd, *82*, *84*, *86*, *171*, *173*, 173, *240*
lsm_l_ent, *146*, 175, *188*, *197*
lsm_l_frac_cv, *87*, *89*, *90*, 176, *179*, *180*, *241*
lsm_l_frac_mn, *87*, *89*, *90*, *177*, 178, *180*, *241*
lsm_l_frac_sd, *87*, *89*, *90*, *177*, *179*, 179, *241*
lsm_l_gyrate_cv, *92*, *94*, *95*, 181, *184*, *185*, *243*
lsm_l_gyrate_mn, *92*, *94*, *95*, *182*, 182, *185*, *243*
lsm_l_gyrate_sd, *92*, *94*, *95*, *182*, *184*, 184, *243*
lsm_l_iji, *97*, 186
lsm_l_joinent, *146*, *176*, 187, *197*

lsm_l_lpi, *98*, 188
lsm_l_lsi, *100*, 190
lsm_l_mesh, *101*, 191
lsm_l_msidi, 193
lsm_l_msiei, 195
lsm_l_mutinf, *146*, *176*, *188*, 196
lsm_l_ndca, *104*, 197, *245*
lsm_l_np, *106*, 200, *208*
lsm_l_pafrac, *108*, 201
lsm_l_para_cv, *109*, *111*, *112*, 203, *205*, *207*, *246*
lsm_l_para_mn, *109*, *111*, *112*, *204*, 204, *207*, *246*
lsm_l_para_sd, *109*, *111*, *112*, *204*, *205*, 206, *246*
lsm_l_pd, *114*, 207
lsm_l_pladj, 209
lsm_l_pr, 210, *212*, *221*, *223*, *224*
lsm_l_prd, 211
lsm_l_rpr, 213
lsm_l_shape_cv, *118*, *120*, *121*, 214, *217*, *218*, *249*
lsm_l_shape_mn, *118*, *120*, *121*, *215*, 216, *218*, *249*
lsm_l_shape_sd, *118*, *120*, *121*, *215*, *217*, 217, *249*
lsm_l_shdi, 219, *222*
lsm_l_shei, 220
lsm_l_sidi, *194*, 222
lsm_l_siei, *196*, 223
lsm_l_split, *123*, 225
lsm_l_ta, *42*, *69*, *71*, *79*, *81*, *98*, *101*, *114*, *117*, *123*, *130*, *161*, *168*, *170*, *190*, *193*, *208*, *226*, 226, *232*
lsm_l_tca, *124*, 228, *238*
lsm_l_te, *126*, *170*, 229, *230*
lsm_p_area, *37*, *39*, *40*, *42*, *56*, *79*, *98*, *101*, *108*, *123*, *129*, *130*, *132*, *144*, *168*, *190*, *193*, *202*, *226*, *227*, 231, *233*, *235*, *241*, *246*, *249*
lsm_p_cai, *44*, *46*, *48*, *134*, *136*, *138*, 232
lsm_p_circle, *50*, *51*, *53*, *140*, *141*, *143*, 234
lsm_p_contig, *57*, *59*, *61*, *149*, *151*, *152*, 235
lsm_p_core, *63*, *65*, *67*, *69*, *124*, *154*, *156*, *158*, *229*, *233*, 237
lsm_p_enn, *82*, *84*, *86*, *171*, *173*, *174*, 239
lsm_p_frac, *87*, *89*, *90*, *177*, *179*, *180*, 240
lsm_p_gyrate, *92*, *94*, *95*, *182*, *184*, *185*, 242
lsm_p_ncore, *73*, *76*, *78*, *104*, *163*, *165*, *167*, *199*, 243
lsm_p_para, *109*, *111*, *112*, *204*, *205*, *207*, 245
lsm_p_perim, *56*, *108*, *126*, *144*, *202*, *230*, *241*, *246*, 247, *249*
lsm_p_shape, *100*, *105*, *118*, *120*, *121*, *191*, *215*, *217*, *218*, 248

mean, *39*, *46*, *50*, *51*, *53*, *65*, *76*, *84*, *89*, *94*, *111*, *120*, *130*, *136*, *140*, *141*, *143*, *156*, *165*, *173*, *179*, *184*, *205*, *217*

options_landscapemetrics, 250

podlasie_ccilc, 250

sample_lsm, 251, *256*
scale_sample, 254, *259*
scale_window, 257
sd, *40*, *48*, *67*, *78*, *86*, *90*, *112*, *121*, *132*, *138*, *158*, *167*, *174*, *180*, *207*, *218*
show_cores, 259
show_correlation, 262
show_lsm, 263, *270*
show_patches, 266
spatialize_lsm, 267
sum, *42*, *227*

window_lsm, *259*, 270