Package ‘vectormetrics’

April 24, 2023

Title Landscape Metrics for Categorical Map Patterns in Vector Data

Version 0.1

Description What the package does (one paragraph).

License GPL-3

Depends R (>= 3.1)

Encoding UTF-8

LazyData true

Imports dplyr,
purrr,
rlang,
sf,
tibble

Suggests sp

RoxygenNote 6.1.1

R topics documented:

`vector_landscape` *Example map (random cluster neutral landscape model).*

Description

An example map to show landscapetools functionality generated with the ‘nlm_randomcluster()‘ algorithm and coerced to vector data (multipolygon).

Usage

```
vector_landscape
```

Format

A sf object.

Source

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

`vm_c_area_cv` *The coefficient of variation of all patch areas at class level (vector data)*

Description

This function allows you to calculate the coefficient of variation of all patch areas belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_area_cv(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated coefficient of variation of areas of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_area_cv(vector_landscape, "class")
```

vm_c_area_mn *The mean value of all patch areas at class level(vector data)*

Description

This function allows you to calculate the mean value of all patch areas belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_area_mn(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated mean value of areas of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_area_mn(vector_landscape, "class")
```

vm_c_area_sd *The standard deviation of all patch areas at class level(vector data)*

Description

This function allows you to calculate the standard deviation of all patch areas belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_area_sd(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated standard deviation of areas of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_area_sd(vector_landscape, "class")
```

vm_c_ca *The total area of each class(vector data)*

Description

This function allows you to calculate the total area of each class in a categorical landscape in vector data format

Usage

```
vm_c_ca(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated total class areas are in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_ca(vector_landscape, "class")
```

---

vm_c_cai_cv *The coefficient of variation of ratio of the core area and the area at class level(vector data)*

Description

This function allows you to calculate the coefficient of variation of the ratio of the core area and the area belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_cai_cv(landscape, class, edge_depth)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape
`edge_depth`	the fixed distance to the edge of the patch

Value

the returned calculated coefficient of variation of ratio of the core area and the area of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_cai_cv(vector_landscape, "class", edge_depth = 1)
```

vm_c_cai_mn *The mean value of core area index at class level(vector data)*

Description

This function allows you to calculate the mean value of ratio of the core area and the area belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_cai_mn(landscape, class, edge_depth)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape
edge_depth	the fixed distance to the edge of the patch

Value

the returned calculated mean value of core area index of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_cai_mn(vector_landscape, "class", edge_depth = 1)
```

vm_c_cai_sd *The standard deviation of core area index of all patches at class level(vector data)*

Description

This function allows you to calculate the standard deviation of all the ratio of the core area and the area belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_cai_sd(landscape, class, edge_depth)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape
edge_depth	the fixed distance to the edge of the patch

Value

the returned calculated standard deviation of ratio of the core area and the area of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_cai_sd(vector_landscape, "class", edge_depth = 1)
```

vm_c_circle_cv *The coefficient of variation of Ratios between the patch area and the smallest circumscribing circle of all patches at class level(vector data)*

Description

This function allows you to calculate the coefficient of variation of ratios of each class in a categorical landscape in vector data format the ratio is the patch area relative to area of the smallest circumscribing circle of the patch

Usage

```
vm_c_circle_cv(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated coefficient of variation in each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_circle_cv(vector_landscape, "class")
```

vm_c_circle_mn *The mean value of Ratios between the patch area and the smallest circumscribing circle of patches at class level(vector data)*

Description

This function allows you to calculate the mean value of ratios of each class in a categorical landscape in vector data format the ratio is the patch area relative to area of the smallest circumscribing circle of the patch

Usage

```
vm_c_circle_mn(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated mean value of ratios of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_circle_mn(vector_landscape, "class")
```

vm_c_circle_sd *The standard deviation of Ratios between the patch area and the smallest circumscribing circle of patches at class level(vector data)*

Description

This function allows you to calculate the standard deviation of ratios of each class in a categorical landscape in vector data format the ratio is the patch area relative to area of the smallest circumscribing circle of the patch

Usage

```
vm_c_circle_sd(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated standard deviation of ratios of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_circle_sd(vector_landscape, "class")
```

vm_c_core_cv *The coefficient of variation of each class core areas(vector data)*

Description

This function allows you to calculate the coefficient of variation of all patch core areas belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_core_cv(landscape, class, edge_depth)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape
edge_depth	the fixed distance to the edge of the patch

Value

the returned calculated coefficient of variation of core areas of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_core_cv(vector_landscape, "class", edge_depth= 1)
```

vm_c_core_mn *The mean value of all core areas in each class(vector data)*

Description

This function allows you to calculate the mean value of all core patch areas belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_core_mn(landscape, class, edge_depth)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape
edge_depth	the fixed distance to the edge of the patch

Value

the returned calculated mean value of core areas of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_core_mn(vector_landscape, "class", edge_depth= 1)
```

vm_c_core_sd *The standard deviation of all core areas of each class(vector data)*

Description

This function allows you to calculate the standard deviation of all patch core areas belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_core_sd(landscape, class, edge_depth)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape
edge_depth	the fixed distance to the edge of the patch

Value

the returned calculated standard deviation of core areas of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_core_sd(vector_landscape, "class", edge_depth= 1)
```

vm_c_cpland *Core area percentage of landscape in each class(vector data)*

Description

This function allows you to calculate the total core area of each class in relation to the landscape area in a categorical landscape in vector data format

Usage

```
vm_c_cpland(landscape, class, edge_depth)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape
`edge_depth`	the fixed distance to the edge of the patch

Value

the returned calculated ratios are in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_cpland(vector_landscape, "class", edge_depth = 1)
```

vm_c_dcad *Disjunct core area density of each class(vector data)*

Description

This function allows you to calculate the number of disjunct core areas of each class in relation to the landscape area in a categorical landscape in vector data format

Usage

```
vm_c_dcad(landscape, class, edge_depth)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape
`edge_depth`	the fixed distance to the edge of the patch

Value

the returned calculated ratios are in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_dcad(vector_landscape, "class", edge_depth = 1)
```

vm_c_dcore_cv *The coefficient of variation of number of disjunct core areas in each class (vector data)*

Description

This function allows you to calculate the coefficient of variation of each class belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_dcore_cv(landscape, class, edge_depth)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape
`edge_depth`	the fixed distance to the edge of the patch

Value

the returned calculated coefficient of variation in each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_dcore_cv(vector_landscape, "class", edge_depth= 1)
```

---

vm_c_dcore_mn *The mean value of number of core areas in each class(vector data)*

Description

This function allows you to calculate the mean value of the total number of disjunct core areas belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_dcore_mn(landscape, class, edge_depth)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape
`edge_depth`	the fixed distance to the edge of the patch

Value

the returned calculated mean value of number of core areas of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_dcore_mn(vector_landscape, "class", edge_depth= 1)
```

vm_c_dcore_sd — *The standard deviation of the number of disjunct core areas of each class(vector data)*

Description

This function allows you to calculate the standard deviation of all number of disjunct core areas belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_dcore_sd(landscape, class, edge_depth)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape
edge_depth	the fixed distance to the edge of the patch

Value

the returned calculated standard deviation of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_dcore_sd(vector_landscape, "class", edge_depth= 1)
```

vm_c_division — *Landscape division index of each class(vector data)*

Description

This function allows you to calculate the Landscape division index of each class in a categorical landscape in vector data format, Landscape division index can somehow reflect the probability that two randomly selected points are not located in the same patch of class i

Usage

```
vm_c_division(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated index is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_division(vector_landscape, "class")
```

---

vm_c_ed *The Edge density of each class(vector data)*

Description

This function allows you to calculate the total length of all patches in class i in a categorical landscape in vector data format

Usage

```
vm_c_ed(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated total length of perimeter is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_ed(vector_landscape, "class")
```

vm_c_enn_cv *The coefficient of variation of Euclidean Nearest-Neighbor Distance of all patches at class level(vector data)*

Description

This function allows you to calculate the coefficient of variation of the Euclidean Nearest-Neighbor Distance among patches of one class in a categorical landscape in vector data format Euclidean Nearest-Neighbor Distance means the distance from a patch edge to the nearest neighbouring patch belonging to the same class.

Usage

```
vm_c_enn_cv(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated coefficient of variation of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_enn_cv(vector_landscape, "class")
```

vm_c_enn_mn *The mean value of the Euclidean Nearest-Neighbor Distance of patches in each class(vector data)*

Description

This function allows you to calculate the mean value of Euclidean Nearest-Neighbor Distance among patches of one class in a categorical landscape in vector data format Euclidean Nearest-Neighbor Distance means the distance from a patch edge to the nearest neighbouring patch belonging to the same class.

Usage

```
vm_c_enn_mn(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated mean value of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_enn_mn(vector_landscape, "class")
```

`vm_c_enn_sd` *The standard deviation of Euclidean Nearest-Neighbor Distance of patches in each class(vector data)*

Description

This function allows you to calculate the standard deviation of Euclidean Nearest-Neighbor Distance among patches of one class in a categorical landscape in vector data format

Usage

```
vm_c_enn_sd(landscape, class)
```

Arguments

`landscape`	the input landscape image, should in "POLYGON" or "MULTIPOLYGON" form.
`class`	the name of the class column of the input landscape

Value

the returned calculated standard deviation of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_enn_sd(vector_landscape, "class")
```

vm_c_frac_cv *The coefficient of variation of fractal dimension index of all patches in each class(vector data)*

Description

This function allows you to calculate the coefficient of variation of fractal dimension index of all patches belonging to class i in a categorical landscape in vector data format The index is based on the patch perimeter and the patch area and describes the patch complexity

Usage

```
vm_c_frac_cv(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated coefficient of variation of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_frac_cv(vector_landscape, "class")
```

vm_c_frac_mn *The mean value of the fractal dimension index of all patches in each class(vector data)*

Description

This function allows you to calculate the mean value of fractal dimension index of all patches belonging to class i in a categorical landscape in vector data format The index is based on the patch perimeter and the patch area and describes the patch complexity

Usage

```
vm_c_frac_mn(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated mean value of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_frac_mn(vector_landscape, "class")
```

---

vm_c_frac_sd *The standard deviation of the fractal dimension index of all patches in each class(vector data)*

Description

This function allows you to calculate the standard deviation of fractal dimension index of all patches belonging to class i in a categorical landscape in vector data format The index is based on the patch perimeter and the patch area and describes the patch complexity

Usage

```
vm_c_frac_sd(landscape, class)
```

Arguments

`landscape`	the input landscape image, should in "POLYGON" or "MULTIPOLYGON" form.
`class`	the name of the class column of the input landscape

Value

the returned calculated standard deviation of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_frac_sd(vector_landscape, "class")
```

vm_c_lpi *Largest patch index(vector data)*

Description

This function allows you to calculate the maximal patch area of each class in relative to total landscape area in a categorical landscape in vector data format

Usage

```
vm_c_lpi(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated index are in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_lpi(vector_landscape, "class")
```

vm_c_lsi *Landscape shape index (vector data)*

Description

This function allows you to calculate the ratio between the actual edge length of class i and the hypothetical minimum edge length of class i in a categorical landscape in vector data format. The minimum edge length equals the edge length if class i would be maximally aggregated

Usage

```
vm_c_lsi(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated index are in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_lsi(vector_landscape, "class")
```

vm_c_mesh *Effective Mesh Size (vector data)*

Description

This function helps to analyse the patch structure the calculate process is, each patch is squared before the sums for each class i are calculated and the sum is standardized by the total landscape area. it is a aggregation metric.

Usage

```
vm_c_mesh(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated values are in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_mesh(vector_landscape, "class")
```

vm_c_ndca *the number of disjunct core area in each class(vector data)*

Description

This function allows you to calculate the number of disjunct core areas of each class in a categorical landscape in vector data format

Usage

```
vm_c_ndca(landscape, class, edge_depth)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape
edge_depth	the fixed distance to the edge of the patch

Value

the returned calculated number of core area of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_ndca(vector_landscape, "class", edge_depth = 1)
```

vm_c_np *the number of patches in each class(vector data)*

Description

This function allows you to calculate the number of patches of each class in a categorical landscape in vector data format

Usage

```
vm_c_np(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated number of patches of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_np(vector_landscape, "class")
```

---

vm_c_pafrac *Perimeter-Area Fractal Dimension(vector data)*

Description

This function allows you to get the result of 2 divided by ß, ß is the slope of the regression of the area against the perimeter in logarithm of each class in a categorical landscape in vector data format

Usage

```
vm_c_pafrac(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated slope is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_pafrac(vector_landscape, "class")
```

vm_c_para_cv *The coefficient of variation of Perimeter-Area ratio index of each class (vector data)*

Description

This function allows you to calculate the coefficient of variation of the ratios of all patches belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_para_cv(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated coefficient of variation of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_para_cv(vector_landscape, "class")
```

vm_c_para_mn *The mean value of all patches Perimeter-Area ratio index at class level(vector data)*

Description

This function allows you to calculate the mean value of all patch ratios belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_para_mn(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated mean value of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_para_mn(vector_landscape, "class")
```

vm_c_para_sd *The standard deviation of Perimeter-Area ratio index of all patches at class level(vector data)*

Description

This function allows you to calculate the standard deviation of the ratios of all patches belonging to one class in a categorical landscape in vector data format

Usage

```
vm_c_para_sd(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated standard deviation of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_area_sd(vector_landscape, "class")
```

vm_c_pd — *the patch density in each class(vector data)*

Description

This metric is based on categorical landscape in vector data format. The density is the number of patches of each class relative to the total landscape area. Then the number is standardised, so that the comparison among different landscape is possible.

Usage

```
vm_c_pd(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated density of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_pd(vector_landscape, "class")
```

vm_c_pland — *Percentage of landscape of class(vector data)*

Description

This function allows you to calculate the percentage of each class in a categorical landscape in vector data format That means each class area is standardised by the total landscape area, so the comparision among different landscape is possible.

Usage

```
vm_c_pland(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated percentage of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_pland(vector_landscape, "class")
```

vm_c_shape_cv *The coefficient of variation of Shape index of each class (vector data)*

Description

This function allows you to calculate the coefficient of variation of shape index of all patches belonging to one class in a categorical landscape in vector data format shape index is 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. That means, the perimeter of a circle with the same area of the patch.

Usage

```
vm_c_shape_cv(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated coefficient of variation of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_shape_cv(vector_landscape, "class")
```

vm_c_shape_mn *The mean value of Shape index of each class (vector data)*

Description

This function allows you to calculate the mean value of shape index of all patches belonging to one class in a categorical landscape in vector data format shape index is 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. That means, the perimeter of a circle with the same area of the patch.

Usage

```
vm_c_shape_mn(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated mean value of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_shape_mn(vector_landscape, "class")
```

vm_c_shape_sd *The standard deviation of Shape index of each class (vector data)*

Description

This function allows you to calculate the standard deviation of shape index of all patches belonging to one class in a categorical landscape in vector data format shape index is 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. That means, the perimeter of a circle with the same area of the patch.

Usage

```
vm_c_shape_sd(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated standard deviation of each class is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_shape_sd(vector_landscape, "class")
```

`vm_c_split` *Splitting index (vector data)*

Description

This function allows you to calculate the relation between square of landscape area and sum of square of all patch area of class i in a categorical landscape in vector data format it is a aggregation metric.

Usage

```
vm_c_split(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated indices are in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_split(vector_landscape, "class")
```

vm_c_tca *The total core area of each class(vector data)*

Description

This function allows you to calculate the total core area of each class in a categorical landscape in vector data format

Usage

```
vm_c_tca(landscape, class, edge_depth)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape
`edge_depth`	the fixed distance to the edge of the patch

Value

the returned calculated total class core areas are in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_tca(vector_landscape, "class", edge_depth= 1)
```

vm_c_te *Total (class) edge (vector data)*

Description

This function allows you to calculate the total length of edge of class i in a categorical landscape in vector data format

Usage

```
vm_c_te(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated length is in column "value", and this function returns also some important information such as level, class number and metric name. Moreover, the "id" column, although it is just NA here at class level. we need it because the output struture of metrics at class level should correspond to patch level one by one, and then it is more convinient to combine metric values at different levels and compare them.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_c_te(vector_landscape, "class")
```

---

vm_l_area_mn *The mean value of all patch areas at landscape level(vector data)*

Description

This function allows you to calculate the mean value of all patch areas in a categorical landscape in vector data format

Usage

```
vm_l_area_mn(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated mean value of areas is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_area_mn(vector_landscape, "class")
```

vm_l_cai_mn *The mean value of core area index at landscape level(vector data)*

Description

This function allows you to calculate the mean value of ratio of the core area and the area in a categorical landscape in vector data format

Usage

```
vm_l_cai_mn(landscape, class, edge_depth)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape
`edge_depth`	the fixed distance to the edge of the patch

Value

the returned calculated mean value of core area index of the whole landscape is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_cai_mn(vector_landscape, "class", edge_depth = 1)
```

vm_l_core_mn *The mean value of all core areas in landscape(vector data)*

Description

This function allows you to calculate the mean value of all core patch areas in a categorical landscape in vector data format

Usage

```
vm_l_core_mn(landscape, class, edge_depth)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape
`edge_depth`	the fixed distance to the edge of the patch

Value

the returned calculated mean value of core area of the whole landscape is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_core_mn(vector_landscape, "class", edge_depth= 1)
```

vm_l_dcad *Disjunct core area density of the whole landscape(vector data)*

Description

This function allows you to calculate the number of disjunct core areas in relation to the landscape area in a categorical landscape in vector data format

Usage

```
vm_l_dcad(landscape, class, edge_depth)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape
edge_depth	the fixed distance to the edge of the patch

Value

the returned calculated ratios are in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_dcad(vector_landscape, "class", edge_depth = 1)
```

vm_l_dcore_mn *The mean value of number of core areas in landscape(vector data)*

Description

This function allows you to calculate the mean value of the total number of disjunct core areas in a categorical landscape in vector data format

Usage

```
vm_l_dcore_mn(landscape, class, edge_depth)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape
`edge_depth`	the fixed distance to the edge of the patch

Value

the returned calculated mean value of number of the whole landscape is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_dcore_mn(vector_landscape, "class", edge_depth= 1)
```

vm_l_division *Landscape division index of the whole landscape(vector data)*

Description

This function allows you to calculate the Landscape division index in a categorical landscape in vector data format, Landscape division index can somehow reflect the probability that two randomly selected points are not located in the same patch

Usage

```
vm_l_division(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated index is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_division(vector_landscape, "class")
```

vm_l_ed *The Edge density of the whole landscape(vector data)*

Description

This function allows you to calculate the total length of all patches in a categorical landscape in vector data format

Usage

```
vm_l_ed(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated total length of perimeter is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_ed(vector_landscape, "class")
```

`vm_l_frac_mn` *The mean value of the fractal dimension index of landscape(vector data)*

Description

This function allows you to calculate the mean value of fractal dimension index in a categorical landscape in vector data format The index is based on the patch perimeter and the patch area and describes the patch complexity

Usage

```
vm_l_frac_mn(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated mean value of the whole landscape is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_frac_mn(vector_landscape, "class")
```

`vm_l_lpi` *Largest patch index(vector data)*

Description

This function allows you to calculate the maximal patch area in relative to total landscape area in a categorical landscape in vector data format

Usage

```
vm_l_lpi(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated index are in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_lpi(vector_landscape, "class")
```

vm_l_lsi *Landscape shape index (vector data)*

Description

This function allows you to calculate the ratio between the actual edge length of class i and the hypothetical minimum edge length in a categorical landscape in vector data format. The minimum edge length equals the edge length if class i would be maximally aggregated

Usage

```
vm_l_lsi(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated index are in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level ## if the class name of input landscape is landcover, ## then write landcover in a double quotation marks as the second parameter. vm_l_lsi(vector_landscape, "class")

vm_l_mesh *Effective Mesh Size (vector data)*

Description

This function helps to analyse the patch structure the calculate process is, each patch is squared before the sums of them are calculated and the sum is standardized by the total landscape area. it is a aggregation metric.

Usage

```
vm_l_mesh(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated values are in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_mesh(vector_landscape, "class")
```

`vm_l_msidi` *Modified Simpson's diversity index (vector data)*

Description

This function allows you to calculate the Modified Simpson's diversity index in a categorical landscape in vector data format, Modified Simpson's diversity index is diversity index

Usage

```
vm_l_msidi(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated index is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_msidi(vector_landscape, "class")
```

vm_l_msiei *Modified Simpson's evenness index (vector data)*

Description

This function allows you to calculate the Modified Simpson's evenness index in a categorical landscape in vector data format, Modified Simpson's evenness index is diversity index

Usage

```
vm_l_msiei(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated index is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_msiei(vector_landscape, "class")
```

vm_l_ndca *the number of disjunct core area in the whole landscape(vector data)*

Description

This function allows you to calculate the number of disjunct core areas in a categorical landscape in vector data format

Usage

```
vm_l_ndca(landscape, class, edge_depth)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape
`edge_depth`	the fixed distance to the edge of the patch

Value

the returned calculated number of core area is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_ndca(vector_landscape, "class", edge_depth = 1)
```

`vm_l_np` *the number of patches in each class(vector data)*

Description

This function allows you to calculate the number of patches of each class in a categorical landscape in vector data format

Usage

```
vm_l_np(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated number of patches of each class is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_np(vector_landscape, "class")
```

vm_l_pafrac *Perimeter-Area Fractal Dimension(vector data)*

Description

This function allows you to get the result of 2 divided by ß, ß is the slope of the regression of the area against the perimeter in logarithm of all patches in a categorical landscape in vector data format

Usage

```
vm_l_pafrac(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated slope is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_pafrac(vector_landscape, "class")
```

vm_l_para_mn *The mean value of all patches Perimeter-Area ratio index at landscape level(vector data)*

Description

This function allows you to calculate the mean value of all patch ratios in a categorical landscape in vector data format

Usage

```
vm_l_para_mn(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated mean value is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_para_mn(vector_landscape, "class")
```

---

vm_l_pd *the patch density in the whole landscape(vector data)*

Description

This metric is based on categorical landscape in vector data format. The density is the number of patches of the whole landscape relative to the total landscape area. Then the number is standardised, so that the comparison among different landscape is possible.

Usage

```
vm_l_pd(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated density of each class is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_pd(vector_landscape, "class")
```

`vm_l_pr` *Patch richness (vector data)*

Description

This function allows you to calculate the Patch richness in a categorical landscape in vector data format, Patch richness index is a simplest diversity index

Usage

```
vm_l_pr(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated index is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_pr(vector_landscape, "class")
```

`vm_l_prd` *Patch richness density (vector data)*

Description

This function allows you to calculate the Patch richness density in a categorical landscape in vector data format, Patch richness density is diversity index

Usage

```
vm_l_prd(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated index is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_prd(vector_landscape, "class")
```

vm_l_rpr *Relative patch richness (vector data)*

Description

This function allows you to calculate the Relative patch richness in a categorical landscape in vector data format, Relative patch richness is diversity index

Usage

```
vm_l_rpr(landscape, class, class_max)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape
`class_max`	the maximal number of class in your input landscape image

Value

the returned calculated index is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_rpr(vector_landscape, "class")
```

vm_l_shape_mn *The mean value of Shape index of landscape (vector data)*

Description

This function allows you to calculate the mean value of shape index of all patches in a categorical landscape in vector data format shape index is 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. That means, the perimeter of a circle with the same area of the patch.

Usage

```
vm_l_shape_mn(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated mean value of each class is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_shape_mn(vector_landscape, "class")
```

`vm_l_shdi` *Shannon's diversity index (vector data)*

Description

This function allows you to calculate the Shannon's diversity index in a categorical landscape in vector data format, Shannon's diversity index is diversity index

Usage

```
vm_l_shdi(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated index is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_shdi(vector_landscape, "class")
```

vm_l_shei *Shannons's evenness index (vector data)*

Description

This function allows you to calculate the Shannons's evenness index in a categorical landscape in vector data format, Shannons's evenness index is diversity index It is the ratio between the actual Shannon's diversity index and and the theoretical maximum of the Shannon diversity index

Usage

```
vm_l_shei(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated index is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_shei(vector_landscape, "class")
```

vm_l_sidi *Simpson's diversity index (vector data)*

Description

This function allows you to calculate the Simpson's diversity index in a categorical landscape in vector data format, Simpson's diversity index is diversity index

Usage

```
vm_l_sidi(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated index is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_sidi(vector_landscape, "class")
```

---

vm_l_siei *Simpson's evenness index (vector data)*

Description

This function allows you to calculate the Simpson's evenness index in a categorical landscape in vector data format, Simpson's evenness index is diversity index. It is the ratio between the actual Simpson's diversity index and the theoretical maximum Simpson's diversity index

Usage

```
vm_l_siei(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated index is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_sidi(vector_landscape, "class")
```

`vm_l_split` *Splitting index (vector data)*

Description

This function allows you to calculate the relation between square of landscape area and sum of square of all patch area in a categorical landscape in vector data format it is a aggregation metric.

Usage

```
vm_l_split(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated indices are in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_split(vector_landscape, "class")
```

`vm_l_ta` *The total area of the whole landscape(vector data)*

Description

This function allows you to calculate the total area of the whole landscape in a categorical landscape in vector data format

Usage

```
vm_l_ta(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated total class areas are in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_ta(vector_landscape, "class")
```

vm_l_tca *The total core area of the whole landscape(vector data)*

Description

This function allows you to calculate the total core area in a categorical landscape in vector data format

Usage

```
vm_l_tca(landscape, class, edge_depth)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape
edge_depth	the fixed distance to the edge of the patch

Value

the returned calculated total class core areas are in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_tca(vector_landscape, "class", edge_depth= 1)
```

vm_l_te *Total (class) edge(vector data)*

Description

This function allows you to calculate the total length of edge of whole landscape in a categorical landscape in vector data format

Usage

```
vm_l_te(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated length is in column "value", and this function returns also some important information such as level and metric name, Moreover, class number and the "id" column, although both are "NA" here in the landscape level

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_l_te(vector_landscape, "class")
```

`vm_p_area` *AREA (patch level)*

Description

Patch area (Area and edge metric)

Usage

```
vm_p_area(landscape, class)
```

Arguments

`landscape`	sf* object.
`class`	Column in sf* object indicating the land use type

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/frags

Examples

```
vm_p_area(vector_landscape, "class")
```

---

vm_p_cai *core area index(vector data)*

Description

This function allows you to calculate the ratio of the core area and the area in square meters. Core area is defined as an area that within the patch and its edge is a fixed value from the boundary of the patch. The index describes the percentage of a patch that is core area.

Usage

```
vm_p_cai(landscape, class, edge_depth)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape
`edge_depth`	the fixed distance to the edge of the patch

Value

the returned calculated indices of all patches are in column "value", and this function returns also some important information such as level, class, patch id and metric name.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_p_cai(vector_landscape, "class", edge_depth = 0.8)
```

vm_p_circle *Ratio between the patch area and the smallest circumscribing circle of the patch.*

Description

This function allows you to calculate the ratio between the patch area and the smallest circumscribing circle of the patch. The diameter of the smallest circumscribing circle is the The distance between the two farthest points on the patch.

Usage

```
vm_p_circle(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the function returns the calculated ratio of all patches in column "value", and this function returns also some important information such as level, class, patch id and metric name.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_p_circle(vector_landscape, "class")
```

vm_p_core *core area(vector data)*

Description

This function allows you to calculate the core area of all patches in square meters Core area is defined as an area that within the patch and its edge is a fixed value from the boundary of the patch.

Usage

```
vm_p_core(landscape, class, edge_depth)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape
`edge_depth`	the fixed distance to the edge of the patch

Value

the returned calculated core areas of all patches are in column "value", and this function returns also some important information such as level, class, patch id and metric name.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_p_core(vector_landscape, "class", edge_depth = 0.8)
```

vm_p_enn *Euclidean Nearest-Neighbor Distance(vector data)*

Description

This function allows you to calculate the distance to the nearest neighbouring patch of the same class in meters The distance is measured from edge-to-edge.

Usage

```
vm_p_enn(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated distances of all patches are in column "value", and this function returns also some important information such as level, class, patch id and metric name.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_p_enn(vector_landscape, "class")
```

vm_p_frac *fractal dimension index(vector data)*

Description

This function allows you to calculate index fractal dimension index. The index is based on the patch perimeter and the patch area and describes the patch complexity.

Usage

```
vm_p_frac(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated indice of all patches are in column "value", and this function returns also some important information such as level, class, patch id and metric name.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_p_frac(vector_landscape, "class")
```

`vm_p_ncore` *the number of disjunct core area(vector data)*

Description

This function allows you to calculate the number of disjunct core areas of all patches Core area is defined as an area that within the patch and its edge is a fixed value from the boundary of the patch. Disjunct core area is defined as a new discrete area(patch), which is a sub-part of core area

Usage

```
vm_p_ncore(landscape, class, edge_depth)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape
`edge_depth`	the fixed distance to the edge of the patch

Value

the returned calculated core areasnumber of disjunct core areas of all patches are in column "value", and this function returns also some important information such as level, class, patch id and metric name.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_p_core(vector_landscape, "class", edge_depth = 0.8)
```

vm_p_para *Perimeter-Area ratio.*

Description

This function allows you to calculate the ratio between the patch area and perimeter. The ratio describes the patch complexity in a straightforward way.

Usage

```
vm_p_para(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the function returns the calculated ratio of all patches in column "value", and this function returns also some important information such as level, class, patch id and metric name.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_p_para(vector_landscape, "class")
```

vm_p_perim *The perimeter of each patches(vector data)*

Description

This function allows you to calculate the perimeter of each patches in a categorical landscape in vector data format

Usage

```
vm_p_perim(landscape, class)
```

Arguments

landscape	the input landscape image,
class	the name of the class column of the input landscape

Value

the returned calculated perimeter of all patches is in column "value", and this function returns also some important information such as level, class, patch id and metric name. ## if the class name of input landscape is landcover, ## then write landcover in a double quotation marks as the second parameter. st_p_area(vector_landscape, "class")

`vm_p_shape` *Shape index(vector data)*

Description

This function allows you to calculate the shape index, which is 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. That means, the perimeter of a circle with the same area of the patch.

Usage

```
vm_p_shape(landscape, class)
```

Arguments

`landscape`	the input landscape image,
`class`	the name of the class column of the input landscape

Value

the returned calculated indices of all patches are in column "value", and this function returns also some important information such as level, class, patch id and metric name.

Examples

```
## if the class name of input landscape is landcover,
## then write landcover in a double quotation marks as the second parameter.
vm_p_shape(vector_landscape, "class")
```

Index